Introducing `selfisher`: open source software for statistical analyses of fishing gear selectivity

Authors and affiliations:

Mollie E Brooks[a] , Valentina Melli[a], Esther Savina[a], Juan Santos[b], Russell Millar[c], Finbarr Gerard O'Neill[a] , Tiago Veiga-Malta[a] , Ludvig Ahm Krag[a], Jordan Paul Feekings[a]

[a] National Institute of Aquatic Resources, Technical University of Denmark, Lyngby, 2800 Denmark

[b] Thünen Institute of Baltic Sea Fisheries, Rostock, Germany

[c] Department of Statistics, The University of Auckland, Private Bag 92019, Auckland, New Zealand

Keywords: catch comparison, covered codend, gillnet, mesh size, paired gear, trawl

Running Headline: R package `selfisher` for gear selectivity

Abstract

Fishing gear is constantly being improved to select certain sizes and species while excluding others. Experiments are conducted to quantify the selectivity and the resulting data needs to be analyzed using specialized statistical methods in many cases. Here, we present a new estimation tool for analyzing this type of data: an R package named `selfisher`. It can be used for both active and passive gears, and can handle different trial designs. It allows fitting models containing multiple fixed effects (e.g. length, total catch weight, mesh size, water turbidity) and random effects (e.g. haul). A bootstrapping procedure is provided to account for between and within haul variability and overdispersion. We demonstrate its use via four case studies including (1) covered codend analyses of four gears, (2) a paired gear study with numerous potential covariates, (3) a catch comparison study of unpaired hauls of gillnets and (4) a catch comparison study of paired hauls using polynomials and splines. This free and open source software will make it easier to model fishing gear selectivity, teach the statistical methods, and make analyses more repeatable.

Table of contents

Introduction

Fisheries aim to select for certain species and sizes of individuals while allowing others to avoid capture. Experiments are conducted to measure the selectivity of fishing gear and statistical models are used to characterize the selectivity patterns. The selectivity of fishing gear is commonly described by a retention curve, i.e. the probability of being retained in the net, which is usually a function of individual length or size and may vary between hauls (Wileman et al., 1996). Between-haul variation may be random, or it may depend on observed covariates such as total catch weight (Fryer, 1991; Suuronen & Millar, 1992; Erickson et al., 1996; O'Neill & Kynoch, 1996), environmental variables (He, 1993; Walsh & Hickey, 1993; Ryer & Barnett, 2006, Somerton et al., 2013), or the condition of individuals (Özbilgin et al., 2007; Ferro et al., 2008).

In some cases, catch data collected in selectivity studies could be analyzed with logistic regression methods, i.e. binomial generalized linear models, for which there are plenty of software options available. Binomial models are appropriate because, in many gear selectivity experiments, individuals end up in one of two compartments (e.g. codend vs cover; gear 1 vs gear 2; or test gear vs control gear; Wileman et al., 1996), i.e. there are two possible outcomes as in coin flips. However, a substantial amount of the analyses in this field are specialized and require specialized software. For example, obtaining confidence intervals on predictions typically requires accounting for extra-binomial variability (overdispersion) between and within hauls (Millar, 1993; Millar et al., 2004). Paired gear studies (where the test gear is tested against a control one with retention probability equal to one for the given species and lengths of interest) is another example that does not conform to a typical logistic regression model because the probability model is more complex as we will show below. This paper presents newly developed open source software that is specifically designed for modelling fishing gear selectivity, something that was previously limited. The package, `selfisher`, is implemented in the R statistical computing environment which is commonly used for many modern fisheries analyses (R core team 2020). The package is written with an interface that will be familiar to many users of regression methods in R. By making the software free and openly available, we aim to improve repeatability of analyses and enable teaching these analytical methods in classrooms for the next generation of fisheries scientists.

In this paper, first, we briefly describe the implementation of the `selfisher` R package. We describe how to install the latest version of the package and where to report bugs and where to reach out to other users with questions. Then, we provide a general description of how models are estimated by the package, including dealing with subsampled catches, a common occurrence in gear selectivity studies (e.g. Larsen et al., 2018; Melli et al., 2019; Veiga-Malta et al., 2020). We also address the common issue of overdispersion related to between-haul variability (Millar et al., 2004). We describe three general categories of statistical models, divided based on the mathematical probabilities underlying the estimation, while omitting details about experimental designs and code to run the models. Then, we describe the bootstrapping procedure used to account for variation within and between hauls in selectivity (Millar, 1993), potentially resampling from distinct pools of hauls based on gear type or tactic as in unpaired hauls (Herrmann et al., 2017; Savina et al., 2017). Then, we briefly describe four case studies used to illustrate the package's capabilities, while providing more thorough descriptions with R code in online appendices. The case studies are (1) a covered codend study of haddock (*Melanogrammus aeglefinus*) with four different codends (O'Neill et al., 2016), (2) a paired gear study in a brown shrimp (*Crangon crangon*) trawl fishery where one trawl is nonselective (Santos et al., 2018), (3) a catch comparison study of unpaired hauls of gillnets avoiding an unwanted crab (*Cancer pagurus*) (Savina et al., 2017), and (4) a catch comparison study of paired hauls in a Norway lobster (*Nephrops norvegicus*) trawl fishery (Melli et al., 2018). Finally, we discuss current limitations of the `selfisher` package and potential future advancements.

Implementation of selfisher

The selfisher package was designed to be flexible and robust for fitting and assessing a variety of gear selectivity models that can be represented with a binomial distribution. The code for selfisher was developed by modifying the R package glmmTMB (Brooks et al., 2017) because glmmTMB already had the capabilities needed for fitting and analyzing binomial mixed effect models. Previously, glmmTMB was developed by adapting the popular user interface from lme4 (Bates et al., 2015) and increasing the model flexibility and fitting robustness by doing estimation with TMB (i.e. Template Model Builder). Prior to the development of glmmTMB, TMB was developed based on the algorithm of AD Model Builder (ADMB), which performs maximum likelihood estimation (MLE) in a fast and robust way (Fournier et al., 2012; Miller, 2013; Kristensen et al., 2016). The algorithm is fast and robust because it has information on the gradients of the likelihood surface via automatic differentiation. Additionally, TMB improves robustness by providing binomial and beta-binomial likelihood functions that are numerically stable even when probabilities are near zero or one. Thus, through inheritance, selfisher has a flexible user interface with lme4-style syntax and robust TMB code underlying the model estimation which is done using the same MLE algorithm as ADMB.

Installing selfisher and where to ask questions

The package is continuously being improved and the most recent version can be found online in a GitHub repository. The current version of the package is 1.0.0. News about changes in each version can be found by typing news(package="selfisher")in an R console after installing the package. The package is mature enough that we do not expect to have changes that will affect existing models; most changes will be additions of new features (see Discussion below). The address of the GitHub repository is https://github.com/mebrooks/selfisher. There, you can find installation instructions and a forum for reporting bugs (i.e. issue tracker). Users are discouraged from posting questions on the issue tracker, which is reserved for bugs. For questions about selfisher, an email group is provided where users are encouraged to openly ask and answer each other's questions (https://groups.google.com/d/forum/selfisher-users).

Background of selectivity

All models in `selfisher` involve comparing the catches from two compartments (e.g. test vs control gear, gear 1 vs gear 2, or codend vs cover), which gives rise to data that can be analysed as a binomial response, subject to the use of appropriate methods to allow for overdispersion as described below. Due to technicalities of the underlying code, in `selfisher` syntax, the binomial response must be specified as a proportion (i.e. proportion of individuals of a given length in one compartment of one haul with respect to the total) and a total (i.e. total number of individuals of a given length in either compartment in one haul), as we will demonstrate in case studies.

There are three main categories of experimental designs in which each individual has two possible outcomes, i.e. studies that produce binomial data. The categories are (1) a selective net inside an outer nonselective small-mesh cover net (covered codend), (2) a selective net compared to a nonselective net (paired gear), and (3) a comparison between two selective gears (catch comparison). These can all be modeled using `selfisher` as we describe in sections below, but first we describe some generalities.

Retention models

All three categories of analyses involve estimating a retention model; covered codend and paired gears experiments allow one to estimate the absolute retention, i.e. retention out of the population encountered by the gear, while in a catch comparison experiment the estimated retention is relative to that of a baseline gear. Regardless, the mathematical formulation is general. We use *r(l)* to refer to the retention model throughout this text, but as we describe in case studies below, it may depend on covariates other than length, *l*. See Table 1 for a description of all notations. As in binomial generalized linear models (GLMs), retention models use a link function to keep the retention probability in the range from zero to one. The most common link is the "`logit`" (i.e. logistic), but other options include, "`probit`" (i.e. normal probability ogiv), "`cloglog`" (i.e. negative extreme value), "`loglog`" (i.e. extreme value/Gompertz), or "`Richards`". The software default is the logit link. To fit retention model shapes that are more diverse than the built-in link functions, it is possible to use a logit link with more complex models such as polynomials (Holst & Revill, 2009) or smooth functions (Skalski & Perez-Comas,1993; Munro & Somerton, 2001; Fryer et al., 2003; Somerton et al., 2013) as demonstrated in case studies 3 and 4 below.

Selectivity statistics l_{50} and SR

Two estimated summary statistics of interest are the length with 50% probability of retention (l_{50}) and the selection range (SR, i.e. the width of the range of length classes with 25% to 75% retention probabilities). Note that they only apply to models where retention probability monotonically increases with length, such as in covered codend and paired gear studies. In simple models with only length or size as a predictor of retention, then l_{50} and SR can be extracted from a model using the function `L50SR()`. In more complex models, such as the covered codend and paired gear case studies below, which involve additional covariates besides length, there are multiple ways to extract l_{50} and SR estimates. The covered codend case study demonstrates how to extract them algorithmically by finding the lengths that correspond to retention probabilities 0.25, 0.5, and 0.75 for each given value of covariates in the model. The paired gear example solves for l_{50} and SR mathematically using a model's estimated coefficients. For either method, confidence intervals can be obtained by bootstrapping. In the future, a function will be added to `selfisher` which will be similar to the method demonstrated in the covered codend study.

Subsampled catches

Often, in cases of abundant catches, it may not be feasible to measure the length of every individual that is caught and in those cases, only a fraction of the catch may be measured. This leads to additional statistical complexity in the analyses, but we have made `selfisher` capable of handling subsampling in any model. Here we denote the approximate fraction of individuals in compartment *i*, of haul *h*, and length class *l* that were sampled as $s_{i,h,l}$. Although subsampling doesn't always depend on *l*, we have written the package to be flexible enough to handle cases where each observed count (e.g. $n_{i,h,l}$) has a different subsampling fraction. It is sufficient to include the ratio of subsampling fractions (i.e. the subsampling ratio) in a model, rather than each compartment's fraction individually, $q_{h,l} = s_{i,h,l}/s_{j,h,l}$, assuming here that *i* is the compartment being considered as "success" in the binomial context and *j* is the alternative compartment. If raising factors were recorded in the data instead of subsampling fractions, then care should be taken when calculating the subsampling ratio to account for the fact that a raising factor is the inverse ($1/s_{i,h,l}$) of a subsampling fraction. Subsampling fractions are between zero and one, while raising factors are greater than or equal to one.

The most general way to account for subsampling in a selfisher model is to specify the subsampling ratio $q_{h,l}$ using the argument `qratio` in a call to the `selfisher` function. In covered codend models with logit links (Millar, 1994) or catch comparison models with logit links (Holst & Revill, 2009) an `offset` could be used instead, but using `qratio` will be more broadly applicable because it can be used with any type of link and in paired gear models in addition to the other types.

Three categories of experimental designs

Covered codend

One way of characterizing the selectivity of towed gear is to capture the individuals that escape the net using a small-mesh cover, commonly known as the covered codend method. Then, the statistical model has strong information on retention because the total number of individuals encountered in each length class is directly observed. In this category, we compare the number of individuals sampled in the cover in haul *h* with length *l* ($n_{1,h,l}$) and codend ($n_{2,h,l}$) by modeling the proportion $\frac{n_{1,h,l}}{n_{1,h,l}+n_{2,h,l}}$ as the probability of being retained and sampled in the codend divided by the probability of being retained and sampled in the codend or escaping to the cover and being sampled (Millar, 1994):

$$\phi_{covered,h,l} = \frac{r(l)s_{1,h,l}}{r(l)s_{1,h,l} + (1-r(l))s_{2,h,l}}$$ or more simply in terms of the subsampling ratio:

$$\phi_{covered,h,l} = \frac{r(l)q_{h,l}}{r(l)q_{h,l} + (1-r(l))}.$$

See Table 1 for definitions of all symbols.

Paired gear (where one gear is nonselective)

Another way to characterize the selectivity of a fishing gear is to acquire knowledge on the population available to be caught in each haul. Paired gear studies accomplish this by deploying a control gear or codend (besides the one whose selectivity is being measured) that has full retention for the species and lengths of interest, i.e. all individuals of the given species entering that gear are retained. We assume that there is a probability, p_h , of entering the test gear in haul *h*, given that the individual goes into either the test (*t*) or control (*c*) gear. The proportion of individuals observed in the test gear in haul *h* with length class *l* compared to the total number of individuals observed $\frac{n_{t,h,l}}{n_{t,h,l}+n_{c,h,l}}$ is modeled as the probability of entering the test net, being retained, and being counted, divided by the probability of entering either net, being retained and being counted:

$$\phi_{paired,h,l} = \frac{p_h s_{t,h,l} r(l)}{p_h s_{t,h,l} r(l) + 1 - p_h s_{c,h,l}}$$

It’s convenient to write the model more simply in terms of the subsampling ratio:

$$\phi_{paired,h,l} = \frac{p_h q_{h,l} r(l)}{p_h q_{h,l} r(l) + 1 - p_h}$$

In paired gear studies, the ideal relative fishing power is 50% (i.e. *p*=0.5). If this is known *a priori* then it is possible to fix *p* at 0.5 by specifying `pformula=~0` in the `selfisher` model. If another value of *p* is known *a priori* due to differences in effort, such as swept areas, then that can be specified using the `qratio` argument. For example, if the study has (haul-specific) subsampling fractions *st* and *sc* as well as (haul-specific) swept areas of *at* and *ac* for the test and control gear respectively, then one could use argument `qratio = at / ac * st / sc` together with `pformula=~0` (e.g. Somerton et al., 2013).

Catch comparison (where both gears are selective)

Catch comparison studies compare the catches in two gears, both of which are selective. Consequently, there is no direct information about the length distribution of the population being fished and it is only possible to model the relative retention probability given the population encountered during testing (Revill & Holst, 2004). In general, the relative retention probability model can be of arbitrary shape and, for example, may not be monotone.

The response variable is the proportion of fish retained by one gear versus the other

$$\frac{n_{1,h,l}}{n_{1,h,l}+n_{2,h,l}}\ .$$

In a catch comparison study, the expected proportion of the total catch retained and sampled in gear 1 versus gear 2 is the probability of entering gear 1, being retained in gear 1, and being sampled over the probability of entering, being retained, and being sampled in either gear. It is always modeled with a logit link:

$$\phi_{compare,h,l} = \frac{p_h s_{1,h,l} r_1(l)}{p_h s_{1,h,l} r_1(l) + (1-p_h) s_{2,h,l} r_2(l)}$$

where $r_1(l)$ and $r_2(l)$ are the absolute retention probabilities of the two gears, but it isn't possible to estimate them separately. Holst & Revill (2009) showed that the expected proportion can be approximated with a polynomial with a logit link and an offset to account for any subsampling. They showed that the relative fishing pressure p_h can be absorbed by the intercept and that it may vary randomly between hauls. The `selfisher` package allows fitting a random intercept of haul as in a mixed effects model. In general, the relative retention model in a catch comparison analysis can be formulated as either a polynomial (Holst & Revill, 2009) or a spline (Miller, 2013); see the catch comparison case studies in Appendices 3 and 4 for a demonstration of how this can be done with `selfisher`.

Handling extra variability

Overdispersion

Overdispersion is the presence of variability in the proportions that is in excess of the variability specified under the binomial model. Overdispersion can arise due to between-haul variability whereby the retention model varies from haul to haul. To a lesser extent it can also arise due to within-haul variability due to schooling behavior that violates the assumption that all fish behave independently. The accepted approaches for including this variability are the use of bootstrapping (Millar, 1993) and the use of mixed effects models (Millar et al., 2004). Both of these methods are available in `selfisher`. See the covered codend and paired gear case studies (Appendices 1 and 2) for examples of using mixed effects in `selfisher`. In this text, we do not get into the details of random effects because it is a large topic; however, note that in `selfisher`, they are implemented in the same way as `glmmTMB` (Brooks et al., 2017) using the Laplace approximation, which is a standard method commonly used in modern mixed modelling. All of the case studies demonstrate how to bootstrap as described below.

Quantify uncertainty by bootstrapping

For any statistical model, it is important to compare predictions from the model - and the uncertainty around those predictions - with observed data to ensure that the model reasonably represents the data. A bootstrapping procedure was developed by Millar (1993) to account for variation between and within hauls and calculate appropriately wide confidence intervals. The bootstrapping method can account for overdispersion in data due to variability among hauls (as described above in the *Overdispersion* section); because of this, it is not necessary to include a random effect of haul in models to be bootstrapped. The bootstrapping method is also sometimes referred to as a “double bootstrap” in fishing gear selectivity literature, but this term has another meaning in statistics (e.g. Kuk, 1989). This method first resamples the same number of hauls from the observed set, with replacement. Then for each resampled haul, the method resamples observed fish within the haul. Then it refits the model to the resampled data set and the refit model is used to produce values such as predictions or parameter estimates. It is typically repeated one thousand times or more. This bootstrapping method is implemented in `selfisher` in a way that maintains all variables associated with each observed data point, not just length class, e.g. sampling fractions or total catch. This is facilitated by specifying the `haul` argument in the `selfisher` model fitting function; followed by a call to the `bootSel` function.

It is also possible to do resampling from pools of hauls, so that every bootstrapped dataset has the same number of hauls in each pool as in the original data (Herrmann et al., 2017). That is, if the original dataset has H_A hauls of type A and H_B hauls of type B, then it is possible to bootstrap in a way such that simulated data sets have H_A hauls of type A and H_B hauls of type B. This is done using the `pool` argument to the `selfisher` function. The inner part of the bootstrapping method (resampling fish within hauls) is the same as in the regular bootstrapping method. This is useful when hauls of the two gears or tactics being compared are unpaired. See the gillnet case study in Appendix 3 for an example of specifying pools of hauls.

Bootstrapping and mixed modeling

Mixed modelling is a formal method that takes into account possible sources of variability in the data such as variation between hauls, enabling sound hypothesis testing and model selection. However, fitting mixed models can be computationally intensive. Moreover, the researcher is typically interested in obtaining overall selectivity predictions, rather than at the haul level, because these are relevant to the selectivity applied to the fishery. In that case it is necessary to refit the best candidate model, leaving out any random effects. Bootstrapping can then be used to obtain confidence intervals for estimated quantities such as predicted retention curves or l_{50} and SR. See case studies 1 and 2 for examples.

Case studies

1. Covered codend analyses of four codends catching haddock

This case study uses the haddock data from an experiment that employed the covered codend method to investigate the selective performance of four codends (O'Neill et al., 2016). The codends were made from netting materials with different twine bending stiffnesses and mesh sizes.

We begin the case study by looking at just one gear type to demonstrate different link functions that can be used and to show how to account for subsampling, which in this example varies with length. The default link is the logistic, but we also consider the probit and Richards curves and a spline. Having chosen a model, we bootstrap to estimate 95% confidence intervals for the proportion retained by the codend.

We then analyse all four gear types together and investigate the influence of length, mesh size, bending stiffness and catch size. We assume the principle of geometric similarity (as used by Tokai et al. (1996) to investigate grid selection) and explore a number of models and choose the best fit using Bayesian information criterion (BIC). When choosing the best model, we include a random effect of haul to account for between-haul variation. As in the original publication, we show that selection is dependent on all three parameters. Before bootstrapping, we drop the random effect of haul from the best model because the bootstrapping method accounts for between-haul variability and the random effect would slow it down considerably. We bootstrap to estimate confidence intervals for the proportion retained by each gear and numerically solve for l_{50} and SR dependent on covariates. See Appendix 1 for details and code.

2. Paired gear analyses of codend selectivity dependent on mesh size

This case study draws on a subset of data from the German research project CRANNET (Santos et al., 2018). The experimental method consisted of fishing with two identical beam trawls, simultaneously and in parallel on the same shrimp population. One of the trawls mounted a small-mesh (11 mm) control codend with very limited selectivity, assumed to be nonselective on the range of shrimp lengths available for the trawl. The second trawl mounted a test codend. The subset of data analyzed here consists of catch data from 87 hauls, during which 13 diamond-mesh codends varying in mesh size ranging from 19.1 mm to 36.3 mm were tested. The goal was to model l_{50} and SR as a function of mesh size, and to quantify any effect of two additional haul covariates, sea state and catch weight.

The statistical modeling of selectivity begins with a mixed model to formally assess the effect of mesh size, sea state and catch weight, while controlling for random variation among hauls. *A priori*, the assumption of geometric similarity (that is, l_{50} and SR being proportional to mesh size) was assumed to be the default model, and it is shown that this corresponds to using a `I(length/meshsize)` term in the `selfisher` formula interface for the retention model (Baranov, 1948). The default model was compared to several others and found to be preferred (using BIC), and neither sea state nor catch weight had a significant effect.

Having chosen geometric similarity (with respect to mesh size) as the preferred model, this model was refitted without random effects so as to estimate size selectivity at the population level. Bootstrapping was used to obtain appropriate confidence intervals on l_{50} and SR for any given mesh size.

In addition, this case study demonstrates the use of `psplit=TRUE` (unequal fishing power of the paired codends), the use of sampling ratios, use of the `inits()` function to specify good starting values (because without it some models converged to local minima that didn't make any sense), and shows how l_{50} and SR can be obtained directly from the model fitted by `selfisher`. See Appendix 2 for details and code.

3. Catch comparison analyses of unpaired hauls of gillnets avoiding an unwanted crab

This example deals with data from an experiment originally published by Savina et al. (2017). Two soak tactics (12h at day and 12h at night) were compared in the Danish gillnet plaice fishery to estimate the role that the choice of a soak tactic plays in the catch efficiency of both target and unwanted species. This is a subset of the original dataset (one species, two soak tactics) where we are looking at the unwanted invertebrate edible crab (*Cancer pagurus*).

We use the method developed by Herrmann et al. (2017) which was developed for assessing the effect of changing the gear design on the relative length-dependent catch efficiency. This example is representative of experimental fishing where the catch data obtained for two gears or tactics were not collected in pairs, and can allow for a different number of deployments.

This case study is a typical model for catch comparison of multiple haul data without subsampling using a spline. To get confidence intervals on predictions, we bootstrapped from two pools according to tactic (Night vs Day) using the argument `pool=tactic` to the `bootSel` function. See Appendix 3 for details and code.

4. Catch comparison analyses of paired hauls of *Nephrops* twin-rigged trawls

The example is based on the data from Melli et al. (2018). An anterior gear modification, namely the counter-herding device FLEXSELECT, was tested in a twin-rig configuration, where two identical trawls were towed in parallel. One trawl was equipped with FLEXSELECT, referred to as the test trawl, while the other worked as baseline. The aim of the study was to determine if FLEXSELECT could reduce the fish bycatch in a *Nephrops*-directed fishery. The data used in the example are from haddock, which was found to be strongly affected by the counter-herding device.

Following the steps of the published paper, we conducted a catch comparison analysis, modeling the relative retention as a 4th-order polynomial. In addition, we used a spline with 4 degrees of freedom using the `splines` package and performed model selection to determine if it fitted the data better. Considering that part of the hauls were conducted in day-time and part in night-time, “time” was included in the model as an explanatory variable to determine if the length-based efficiency of FLEXSELECT presented diel differences. We predicted both catch comparison rates and catch ratio with bootstrapped confidence intervals using the `predict` and `bootSel` functions from `selfisher`. See Appendix 4 for details and code.

Discussion

We have introduced an open source R package for estimating fishing gear selectivity of both towed and passive gear, making it easier for anyone to analyze fishing gear selectivity data without writing extensive amounts of code. We have demonstrated its broad applicability in four case studies spanning a range of experimental designs. The case studies have shown that results from `selfisher` are comparable to previously published results and that `selfisher` is more flexible than some methods (e.g. a single model to quantify the effect of changing mesh size on l_{50} and SR). Some of the features of `selfisher` that were demonstrated in the case studies are summarized in Table 2. The case studies aim to demonstrate best practices based on current knowledge. However, this is an active area of research and with a new powerful model fitting tool, best practices may change. Even with (or especially with) a powerful tool, analyses require careful thought and checking of results. For example, in complicated models such as paired gear models which contain two submodels (retention and relative fishing power), it may be necessary to be cautious about identifiability of parameters and local optima encountered during maximum likelihood estimation, but better starting values help avoid those issues as demonstrated in Appendix 2 (Bolker et al., 2013).

We have several advancements for the package either planned for the future or already underway. We plan to add functions to calculate discard ratio indices and indicator functions (e.g. Wienbeck et al., 2014; Santos et al., 2016; Veiga-Malta et al., 2019). We may add a function to facilitate model averaging, although it is currently possible to piece this functionality together with the existing features. We have not tried to fit structured non-monotonic curves (e.g. bell-shaped curves of gillnet absolute selectivity based on geometric similarity, Baranov 1948) with `selfisher,` but we will explore this possibility in the future. We will investigate how to choose starting values of parameters in models that have Richards link, to increase robustness. We plan to implement a general method to extract l_{50} and SR from complex models as demonstrated in case studies 1 and 2. To handle overdispersion more elegantly, we plan to add the option of having a beta-binomial distribution for the response (Miller, 2013). We are already in the process of developing a Shiny app, which will facilitate simple standard analyses without the need for writing code; this will help bridge the gap for scientists or managers with extensive experience in gear development but little experience with R. As an open source package, code developers are encouraged to contribute improvements through GitHub such as those listed here.

Having access to a free and open source software should benefit this field of research in several ways. It allows researchers to share code and thereby foster a community for discussion and repeatability. The free nature of the software will enable researchers and managers with limited budgets - such as those in developing countries - to perform analyses themselves. It gives statistical methods of retention modelling a way into classrooms containing the next generation of fisheries scientists who are already learning modern regression methods as part of a general scientific curriculum.

Acknowledgements

We thank William T. Stockhausen for the trick of fixing p_h to any number in a paired gear analysis. We thank the helpful crews of both research and commercial vessels, as well as the fisheries technicians of the research institutes providing essential support during the fishing trips. We are grateful for funding from the European Maritime and Fisheries Fund (EMFF) and the Ministry of Environment and Food of Denmark (Miljø- og Fødevareministeriet) as part of the projects FastTrack – Sustainable, cost effective and responsive gear solutions under the landing obligation (33112-P-15-013), FastTrack II – Sustainable, cost effective and responsive gear solutions under the landing obligation (33112-P-18-051), and Flexselect - Innovative and Flexible solutions for the Nephrops Fisheries (33113-I-16-068).

ORCID

Mollie E Brooks https://orcid.org/0000-0001-6963-8326

Valentina Melli https://orcid.org/0000-0002-4778-9331

Esther Savina http://orcid.org/0000-0003-4564-5112

Juan Santos https://orcid.org/0000-0003-4277-7734

Russell Millar https://orcid.org/0000-0002-1121-8721

Barry O'Neill http://orcid.org/0000-0002-2797-4548

Tiago Alexandre Matias da Veiga Malta http://orcid.org/0000-0001-8880-3243

Ludvig Ahm Krag https://orcid.org/0000-0002-1531-1499

Jordan Paul Feekings https://orcid.org/0000-0001-9103-7332

Data Availability Statement

Data used in the case studies is available as part of the `selfisher` package on GitHub. See the *Installation* section above as well as the appendices.

References

Baranov, F. I. (1948). Theory and assessment of fishing gear. In Theory of fishing with gillnets. Chap. 7. Pishchepromizdat, Moscow. (Translation from Russian by Ontario Dept of Lands, Maple, Ont., 45 pp.)

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software, 67*(1), 1–48. http://dx.doi.org/10.18637/jss.v067.i01

Bolker, B. M., Gardner, B., Maunder, M., Berg, C.W., Brooks, M., Comita, L., Crone, E., Cubaynes, S., Davies, T., de Valpine, P., Ford, J., Gimenez, O., Kéry, M., Kim, E.J., Lennert−Cody, C., Magnusson, A., Martell, S., Nash, J., Nielsen, A., Regetz, J., Skaug, H. and Zipkin, E. (2013), Strategies for fitting nonlinear ecological models in R, AD

Model Builder, and BUGS. *Methods Ecol Evol*, 4: 501-512. https://doi.org/10.1111/2041-210X.12044

Brooks, M. E., Kristensen, K., van Benthem, K. J., Magnusson, A., Berg, C. W., Nielsen, A., Skaug, H.J., Machler, M., & Bolker, B. M. (2017). glmmTMB balances speed and flexibility among packages for zero-inflated generalized linear mixed modeling. *The R journal*, *9*(2), 378-400. https://doi.org/10.3929/ethz-b-000240890

Erickson, D. L., Perez-Comas, J. A., Pikitch, E. K., & Wallace, J. R. (1996). Effects of catch size and codend type on the escapement of walleye pollock (*Theragra chalcogramma*) from pelagic trawls. *Fisheries Research*, *28*(2), 179-196. https://doi.org/10.1016/0165-7836(96)00497-3

Ferro, R. S. T., Özbilgin, H., & Breen, M. (2008). The potential for optimizing yield from a haddock trawl fishery using seasonal changes in selectivity, population structure and fish condition. *Fisheries research*, *94*(2), 151-159. https://doi.org/10.1016/j.fishres.2008.08.018

Fournier, D. A., Skaug, H. J., Ancheta, J., Ianelli, J., Magnusson, A., Maunder, M. N., Nielsen, A., & Sibert, J. (2012). AD Model Builder: using automatic differentiation for statistical inference of highly parameterized complex nonlinear models. *Optimization Methods and Software*, *27*(2), 233-249. https://doi.org/10.1080/10556788.2011.597854

Fryer, R. J. (1991). A model of between-haul variation in selectivity. *ICES Journal of Marine Science*, *48*(3), 281-290. https://doi.org/10.1093/icesjms/48.3.281

Fryer, R. J., Zuur, A. F., & Graham, N. (2003). Using mixed models to combine smooth size-selection and catch-comparison curves over hauls. *Canadian Journal of Fisheries and Aquatic Sciences*, *60*(4), 448-459. https://doi.org/10.1139/f03-029

He, P. (1993). Swimming speeds of marine fish in relation to fishing gears. *ICES Mar. Sci. Symp*, *196*, 183-189.

Herrmann, B., Sistiaga, M., Rindahl, L., & Tatone, I. (2017). Estimation of the effect of gear design changes on catch efficiency: methodology and a case study for a Spanish longline fishery targeting hake (*Merluccius merluccius*). *Fisheries Research*, *185*, 153-160. https://doi.org/10.1016/j.fishres.2016.09.013

Holst, R., & Revill, A. (2009). A simple statistical method for catch comparison studies. *Fisheries Research*, *95*(2-3), 254-259. https://doi.org/10.1016/j.fishres.2008.09.027

Kristensen, K., Nielsen, A., Berg, C.W., Skaug, H., Bell, B.M. (2016). TMB: Automatic Differentiation and Laplace Approximation. *Journal of Statistical Software*, 70(5), 1-21. http://dx.doi.org/10.18637/jss.v070.i05

Kuk, A. Y. (1989). Double bootstrap estimation of variance under systematic sampling with probability proportional to size. *Journal of Statistical Computation and Simulation*, *31*(2), 73-82.

Larsen, R. B., Herrmann, B., Sistiaga, M., Brinkhof, J., & Grimaldo, E. (2018). Bycatch reduction in the Norwegian deep-water shrimp (*Pandalus borealis*) fishery with a double grid selection system. *Fisheries Research, 208*, 267-273. https://doi.org/10.1016/j.fishres.2018.08.007

Melli, V., Karlsen, J. D., Feekings, J. P., Herrmann, B., & Krag, L. A. (2018). FLEXSELECT: counter-herding device to reduce bycatch in crustacean trawl fisheries. *Canadian Journal of Fisheries and Aquatic Sciences*, *75*(6), 850-860. https://doi.org/10.1139/cjfas-2017-0226

Melli, V., Krag, L. A., Herrmann, B., & Karlsen, J. D. (2019). Can active behaviour stimulators improve fish separation from Nephrops (*Nephrops norvegicus*) in a horizontally divided trawl codend?. *Fisheries Research*, *211*, 282-290. https://doi.org/10.1016/j.fishres.2018.11.027

Millar, R. B. (1993). Incorporation of between−haul variation using bootstrapping and nonparametric estimation of selection curves. *Fishery Bulletin*, *91*, 564-572.

Millar, R. B. (1994). Sampling from trawl gears used in sized selectivity experiments. *ICES Journal of Marine Science*, *51*(3), 293-298. https://doi.org/10.1006/jmsc.1994.1030

Millar, R. B., & Fryer, R. J. (1999). Estimating the size-selection curves of towed gears, traps, nets and hooks. *Reviews in Fish Biology and Fisheries*, *9*(1), 89-116. https://doi.org/10.1023/A:1008838220001

Millar, R. B., Broadhurst, M. K., & Macbeth, W. G. (2004). Modelling between-haul variability in the size selectivity of trawls. *Fisheries Research, 67*(2), 171-181. https://doi.org/10.1016/j.fishres.2003.09.040

Miller, T. (2013). A comparison of hierarchical models for relative catch efficiency based on paired-gear data for US Northwest Atlantic fish stocks. *Can. J. Fish. Aquat. Sci. 70*(9), 1306–1316. https://doi.org/10.1139/cjfas-2013-0136

Munro, P. T., & Somerton, D. A. (2001). Maximum likelihood and non-parametric methods for estimating trawl footrope selectivity. *ICES Journal of Marine Science*, *58*(1), 220-229. https://doi.org/10.1006/jmsc.2000.1004

O'Neill F.G., Kynoch R.J., Blackadder L., Fryer, R.J., Eryaşar A.R., Notti, E., & Sala A. (2016). The influence of twine tenacity, thickness and bending stiffness on codend selectivity. *Fisheries Research, 176*, 94–99. https://doi.org/10.1016/j.fishres.2015.12.012

O'Neill, F. G., & Kynoch, R. J. (1996). The effect of cover mesh size and cod-end catch size on cod-end selectivity. *Fisheries Research*, *28*(3), 291-303. https://doi.org/10.1016/0165-7836(96)00501-2

Özbilgin, H., Tosunoğlu, Z., Tokaç, A., & Metin, G. (2007). Seasonal variation in the trawl codend selectivity of picarel (*Spicare smaris*). *ICES Journal of Marine Science*, *64*(8), 1569-1572. https://doi.org/10.1093/icesjms/fsm115

R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Revill, A.S., & Holst, R. (2004) Reducing discards of North Sea brown shrimp (*C. crangon*) by trawl modification. *Fisheries Research,* 68, 113-122. https://doi.org/10.1016/j.fishres.2004.02.001

Ryer, C. H., & Barnett, L. A. (2006). Influence of illumination and temperature upon flatfish reactivity and herding behavior: potential implications for trawl capture efficiency. *Fisheries Research*, *81*(2-3), 242-250. https://doi.org/10.1016/j.fishres.2006.07.001

Santos, J., Herrmann, B., Mieske, B., Stepputtis, D., Krumme, U., & Nilsson, H. (2016). Reducing flatfish bycatch in roundfish fisheries. *Fisheries Research*, *184*, 64-73. https://doi.org/10.1016/j.fishres.2015.08.025

Santos, J., Herrmann, B., Stepputtis, D., Günther, C., Limmer, B., Mieske, B., and Schultz, S. et al. (2018). Predictive framework for codend size selection of brown shrimp (*Crangon crangon*) in the North Sea beam-trawl fishery. *PLoS ONE* 13(7): e0200464. https://doi.org/10.1371/journal.pone.0200464

Savina, E., Krag, L.A., Frandsen, R.P., & Madsen, N. (2017). Effect of fisher's soak tactic on catch pattern in the Danish gillnet plaice fishery. *Fisheries Research, 196*, 56-65. https://doi.org/10.1016/j.fishres.2017.08.009

Skalski, J. R., & Perez-Comas, J. (1993). Nonparametric maximum likelihood estimation of mesh size selectivity. *Fisheries research*, *18*(3-4), 321-334. https://doi.org/10.1016/0165-7836(93)90160-9

Somerton, David A., Weinberg, Kenneth L., Goodman, Scott E., & Chen, Yong. (2013) Catchability of snow crab (*Chionoecetes opilio*) by the eastern Bering Sea bottom trawl survey estimated using a catch comparison experiment. *Canadian Journal of Fisheries and Aquatic Sciences, 70*(12), 1699-1708. https://doi.org/10.1139/cjfas-2013-0100

Suuronen, P., & Millar, R. B. (1992). Size selectivity of diamond and square mesh codends in pelagic herring trawls: only small herring will notice the difference. *Canadian Journal of Fisheries and Aquatic Sciences*, *49*(10), 2104-2117. https://doi.org/10.1139/f92-234

Tokai, T., Omoto, S., Sato, R., & Matuda, K. (1996). A method of determining selectivity curve of separator grid. *Fisheries Research*, *27*(1-3), 51-60. https://doi.org/10.1016/0165-7836(95)00471-8

Veiga-Malta, T., Feekings, J. P., Frandsen, R. P., Herrmann, B., & Krag, L. A. (2020). Testing a size sorting grid in the brown shrimp (*Crangon Crangon* Linnaeus, 1758) beam trawl fishery. *Fisheries Research, 231*, 105716. https://doi.org/10.1016/j.fishres.2020.105716

Veiga-Malta, T., Feekings, J., Herrmann, B., & Krag, L. A. (2019). Industry-led fishing gear development: Can it facilitate the process?. *Ocean & Coastal Management*, *177*, 148-155. https://doi.org/10.1016/j.ocecoaman.2019.05.009

Walsh, S. J., & Hickey, W. M. (1993). Behavioural reactions of demersal fish to bottom trawls at various light conditions. *ICES Mar. Sci. Symp*,196, 68-76.

Wienbeck, H., Herrmann, B., Feekings, J. P., Stepputtis, D., & Moderhak, W. (2014). A comparative analysis of legislated and modified Baltic Sea trawl codends for simultaneously improving the size selection of cod (*Gadus morhua*) and plaice (*Pleuronectes platessa*). *Fisheries Research*, *150*, 28-37. https://doi.org/10.1016/j.fishres.2013.10.007

Wileman, D., Ferro, R. S. T., Fonteyne, R., & Millar, R. B. (1996). Manual of methods of measuring the selectivity of towed fishing gears. *ICES Cooperative Research Report*, *215*.

Table 1. Symbols

$N_{i,h,l}$	Number of fish caught in compartment *i*, haul *h*, in length class *l*
$n_{i,h,l}$	Number of fish sampled in compartment *i*, haul *h*, in length class *l*
$s_{i,h,l}$	Proportion of fish sampled in compartment *i*, haul *h*, in length class *l*
$q_{h,l} = s_{i,h,l}/s_{j,h,l}$	Subsampling ratio for compartment *i* vs *j* in haul *h*, in length class *l*
$r(l)$	Retention probability as a function of length *l*
p_h	Relative fishing power of test vs control gear in haul *h*
$\phi_{paired,h,l}$	Probability model in a paired gear model
$\phi_{compare,h,l}$	Probability model in a catch comparison model
$\phi_{covered,h,l}$	Probability model in a covered codend model

Table 2. A list of `selfisher` features demonstrated in the four case studies with code in Appendices 1 through 4 (respectively).

Feature	**Demonstrated in case studies**
Random effects	1, 2
Fixed effects other than length or size	1, 2
Splines	3, 4
Residual plot	1
Specifying initial values	2
Model selection via information criteria	1, 2
Link functions other than default logit	1

Appendix 1: Covered codend analyses of four codends catching haddock

4 Dec 2020

This example deals with data from an experiment published by O'Neill et al. (2016) that investigated the selectivity of haddock (*Melanogrammus aeglefinus*) in four codends made from netting materials with different twine bending stiffnesses and mesh sizes. Three of the codends had a nominal mesh size of 120mm and one a nominal mesh size of 130mm. The twine bending stiffness values were in the range 0.64 to 1.1kN mm^2. We label the codends as 120low, 120med, 120high and 130med to reflect their mesh size and bending stiffness (as categorised by the netting manufacturers). As in the original analysis, we show that selection is dependent on both of these parameters and the total codend catch weight.

Preliminaries

```
library(selfisher)
library(plyr) #for aggregating data
library(ggplot2); theme_set(theme_bw())
library(parallel) #for bootstrapping in parallel
library(bbmle) #for AICtab BICtab
library(splines)
library(reshape)
```

Data structure

We load the data each row of which corresponds to the fish of a given length from a given haul. The length (cm), haul number, mesh size (mm), twine bending stiffness ($kNmm^2$), number of fish measured from the codend, codend raising factor, number of fish measured from the cover, cover raising factor, catch size (kg) and codend label are specified respectively.

```
data("coverhaddock")
head(coverhaddock)
```

```
##   Length haul  mesh stiffness codend cod_rf cover cov_rf catch   gear
## 1   10.5   36 119.3      0.69      0      1     0   1.00   617 120low
## 2   11.5   36 119.3      0.69      0      1     0   1.00   617 120low
## 3   12.5   36 119.3      0.69      0      1     0   1.00   617 120low
## 4   13.5   36 119.3      0.69      0      1     1   4.78   617 120low
## 5   14.5   36 119.3      0.69      0      1     4   4.78   617 120low
## 6   15.5   36 119.3      0.69      0      1    12   4.78   617 120low
```

Here we can see that the raising factor varied by length class, which is not a problem in `selfisher`.

```
summary(coverhaddock)
```

```
##      Length           haul            mesh          stiffness
##  Min.   :10.5   Min.   : 6.0   Min.   :119.3   Min.   :0.6400
##  1st Qu.:21.5   1st Qu.:13.0   1st Qu.:119.3   1st Qu.:0.6900
##  Median :33.0   Median :22.5   Median :119.6   Median :0.8000
##  Mean   :33.0   Mean   :22.0   Mean   :122.5   Mean   :0.8087
##  3rd Qu.:44.5   3rd Qu.:31.0   3rd Qu.:129.4   3rd Qu.:0.8000
##  Max.   :55.5   Max.   :39.0   Max.   :129.4   Max.   :1.1000
##      codend           cod_rf          cover            cov_rf
##  Min.   :  0.00   Min.   :1.000   Min.   :  0.00   Min.   : 1.000
##  1st Qu.:  0.00   1st Qu.:1.000   1st Qu.:  0.00   1st Qu.: 1.000
##  Median :  0.00   Median :1.000   Median :  3.00   Median : 2.580
##  Mean   : 11.89   Mean   :1.349   Mean   : 24.82   Mean   : 3.437
##  3rd Qu.:  5.00   3rd Qu.:1.701   3rd Qu.: 38.00   3rd Qu.: 4.097
##  Max.   :156.00   Max.   :2.989   Max.   :272.00   Max.   :59.330
##      catch            gear
##  Min.   :267.0   120high:322
##  1st Qu.:482.0   120low :368
##  Median :535.5   120med :276
##  Mean   :556.7   130med :414
##  3rd Qu.:630.0
##  Max.   :979.0
```

We can also see that all the hauls are contained in one data frame. The data is organized into what is called "long format".

Transforming data

For a model in `selfisher`, we need to convert counts into proportions and totals. Unlike other GLM functions for binomial regression, it is **not** possible to specify the binomial variable as a two-column response variable, e.g. `cbind(N_test, N_cover)`.

Because not all fish in the samples were counted, we will account for this in the model (Millar 1994). The values we need in the model are calculated as `cov_rf/cod_rf`. If instead we had a sampling fraction, we would calculate `qratio = sampling_test/sampling_cover` because a sampling fraction is the inverse of a raising factor. We create a new column in the data with the value of `qratio = cov_rf/cod_rf` for each row. An easy way to compute this value by row is to use the `transform` function.

```
coverhaddock = transform(coverhaddock,
    total = codend + cover,
    prop = codend / (codend+cover),
    qratio = cov_rf / cod_rf)
```

We drop rows of data where no fish were observed because they don't contain any information (i.e. where total = 0). This doesn't affect the model except to allow for bootstrapping later.

```
coverhaddock = subset(coverhaddock, !is.na(prop))
```

Single gear model with covered codend

We will start out with a simple example of one gear type ("120low"). So we need to subset the data.

```
coverhaddock_120low = subset(coverhaddock, gear=="120low")
```

The following is a model for multiple haul data from a covered codend experiment with subsampling. It could be argued that there should be a random effect of haul in the models to account for variation among hauls and avoid pseudoreplication, but we will keep it simple for this first example and only include a fixed effect of length.

```
mod_base = selfisher(prop ~ Length, qratio=qratio, total=total, haul=haul, data=coverhaddock_120low)
```

Extracting residuals and other standard methods

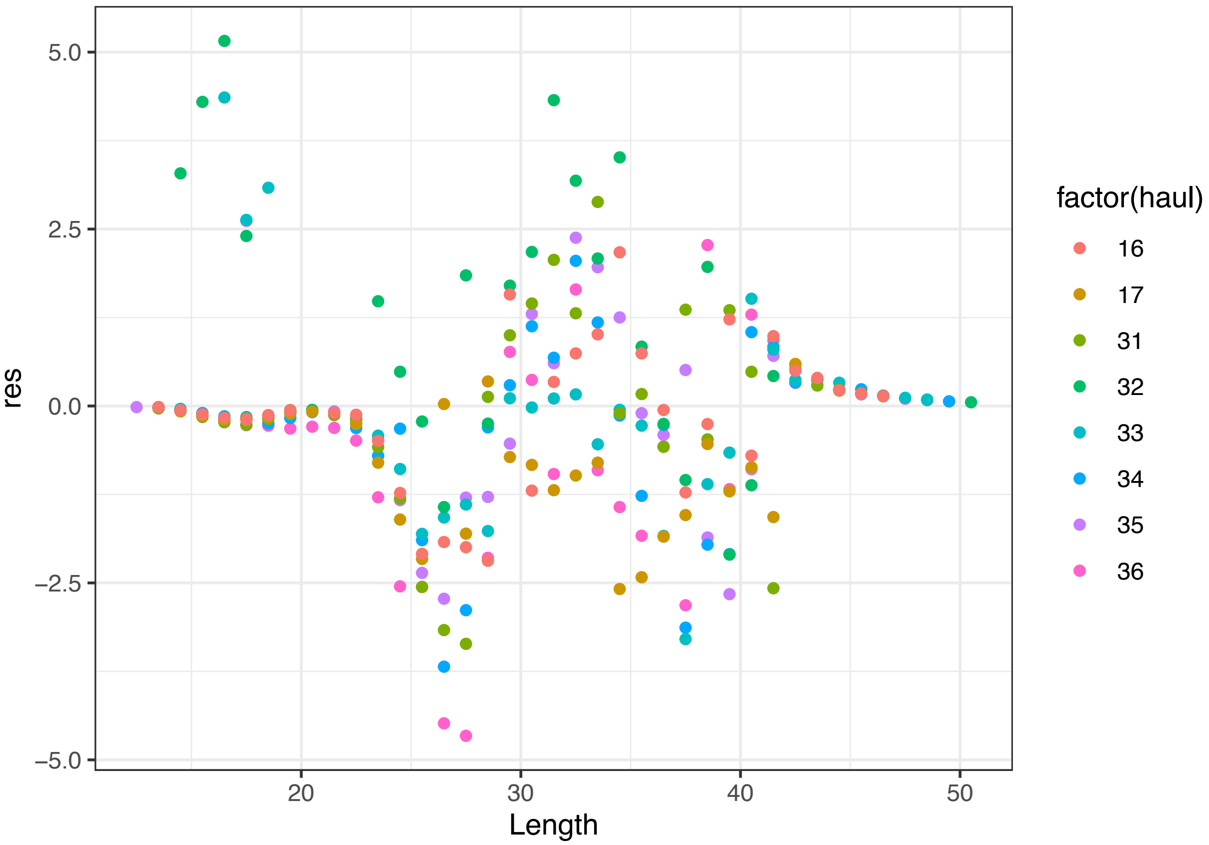
We can check the model residuals for patterns. There's a `residuals` function for `selfisher` models. All methods for `selfisher` models can be displayed as follows:

```
methods(class="selfisher")
```

```
##  [1] anova        confint      df.residual  extractAIC   family       fitted
##  [7] fixef        getME        logLik       model.frame  nobs         predict
## [13] print        ranef        residuals    simulate     summary      terms
## [19] VarCorr      vcov
## see '?methods' for accessing help and source code
```

Here is one way to plot the residuals.

```
coverhaddock_120low$res = residuals(mod_base, type="deviance")
ggplot(coverhaddock_120low, aes(Length, res, colour=factor(haul)))+geom_point()
```



Residuals from GLMs are notoriously opaque, but in the future, we will try to make `selfisher` compatible with the `DHARMa` package to make it easier to assess residuals (Hartig 2020).

Links other than `logit`

It is also possible to consider other link functions, or use a spline. See the function documentation (`?selfisher`) for a list of implemented link functions. Here we fit the logistic, probit and Richard's curve and a spline.

```
mod_probit = selfisher(prop ~ Length, qratio=qratio, total=total,
    link="probit", haul=haul, coverhaddock_120low)

mod_richards = selfisher(prop ~ Length , qratio=qratio, total=total,
    link="richards", haul=haul, coverhaddock_120low)
```

```
## Warning in nlminb(start = par, objective = fn, gradient = gr, control =
## optControl): NA/NaN function evaluation

## Warning in nlminb(start = par, objective = fn, gradient = gr, control =
## optControl): NA/NaN function evaluation
```

```
mod_spline = selfisher(prop ~ bs(Length, 3), qratio=qratio, total=total,
    haul=haul, coverhaddock_120low)
```

Fitting the model with `link="richards"` produced some warnings, but this is ok. The model is valid if the `summary` function is able to produce non-NA standard-errors as seen below.

```
summary(mod_richards)
```

```
##  Family: binomial  ( Richards )
## Selectivity formula:          prop ~ Length
## Data: coverhaddock_120low
## Total: total
##
##          AIC          BIC       logLik     deviance Pearson.ChiSq
##        953.2        963.9       -473.6        491.9       22746.8
##     df.resid
##          255
##
##
## Richards exponent parameter (delta): 0.542
##
## Selectivity model:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -11.65570    0.64953  -17.95   <2e-16 ***
## Length        0.35000    0.01549   22.60   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Size at retention probability:
##      p   Lp.Est Lp.Std.Err
## 1 0.25 32.98060 0.05570708
## 2 0.50 35.54755 0.07331880
## 3 0.75 38.38777 0.15044341
##
## Selectivity range (SR):
##   Estimate Std. Error
##  5.4071721  0.1568771
```

Predictions

To see how the model fits the data, in addition to residuals as above, it helps to plot observations and predictions together. This could be done to examine any of the models above. We could have compared them using log-likelihoods or information criteria, but we'll demonstrate how to do that in the next section.

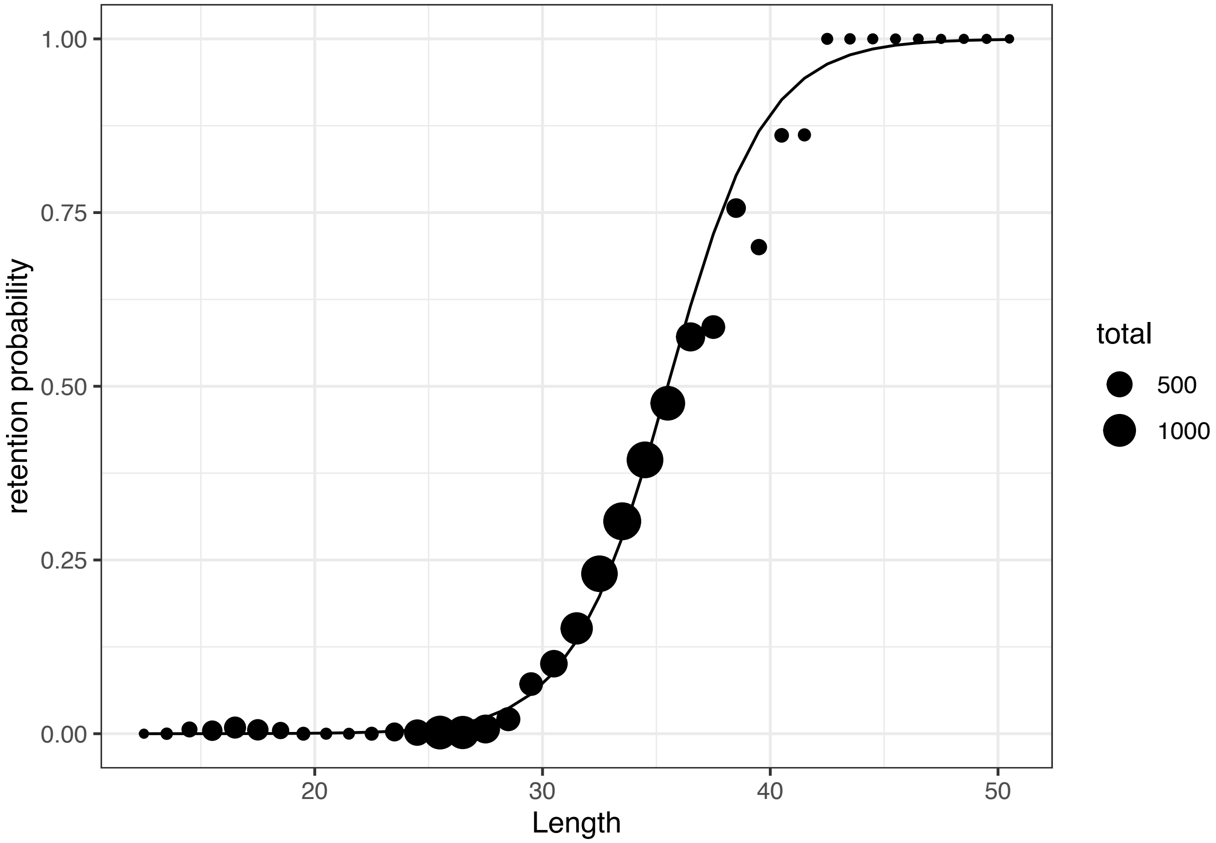
```
newdata = expand.grid(Length=unique(coverhaddock_120low$Length),
    total=1,
    haul=NA,
    qratio=1)
```

```
newdata$prop = predict(mod_base, newdata=newdata, type="response")
```

For plotting observations, we need to aggregate the hauls and raise the data by the raising factor. The raised data is only used for plotting, not for statistical analyses.

```
sumdat_base = ddply(coverhaddock_120low, ~Length, summarize,
    prop = sum(codend)/sum(total),
    total = sum(total),
    raised_prop = sum(codend * cod_rf)/
        sum(codend * cod_rf + cover* cov_rf),
    raised_total=sum(codend * cod_rf + cover* cov_rf)
)
```

```
ggplot(sumdat_base, aes(Length, prop))+
    geom_point(aes(size=total, y=raised_prop))+
    geom_line(data=newdata)+
    ylab("retention probability")
```



Confidence intervals by bootstrapping

Following the method of Millar (1994) the bootstrapping function `bootSel` resamples hauls, then resamples fish within hauls, fits the model to the resampled data, then applies a function `FUN` to each fitted model. In the code below, we define `FUN` to make predictions from each fitted model onto `newdata`. The type of predictions we want in this case are the retention probabilities, i.e. the estimated selection curve, so we specify `type="selection"`. To read about the `predict` function, type `?predict.selfisher` in the R console.

Mac and Linux bootstrapping in parallel

```
bs = bootSel(mod_base, nsim=1000, parallel="multicore", ncpus=4,
    FUN=function(mod){predict(mod, newdata=newdata, type="selection")})
```

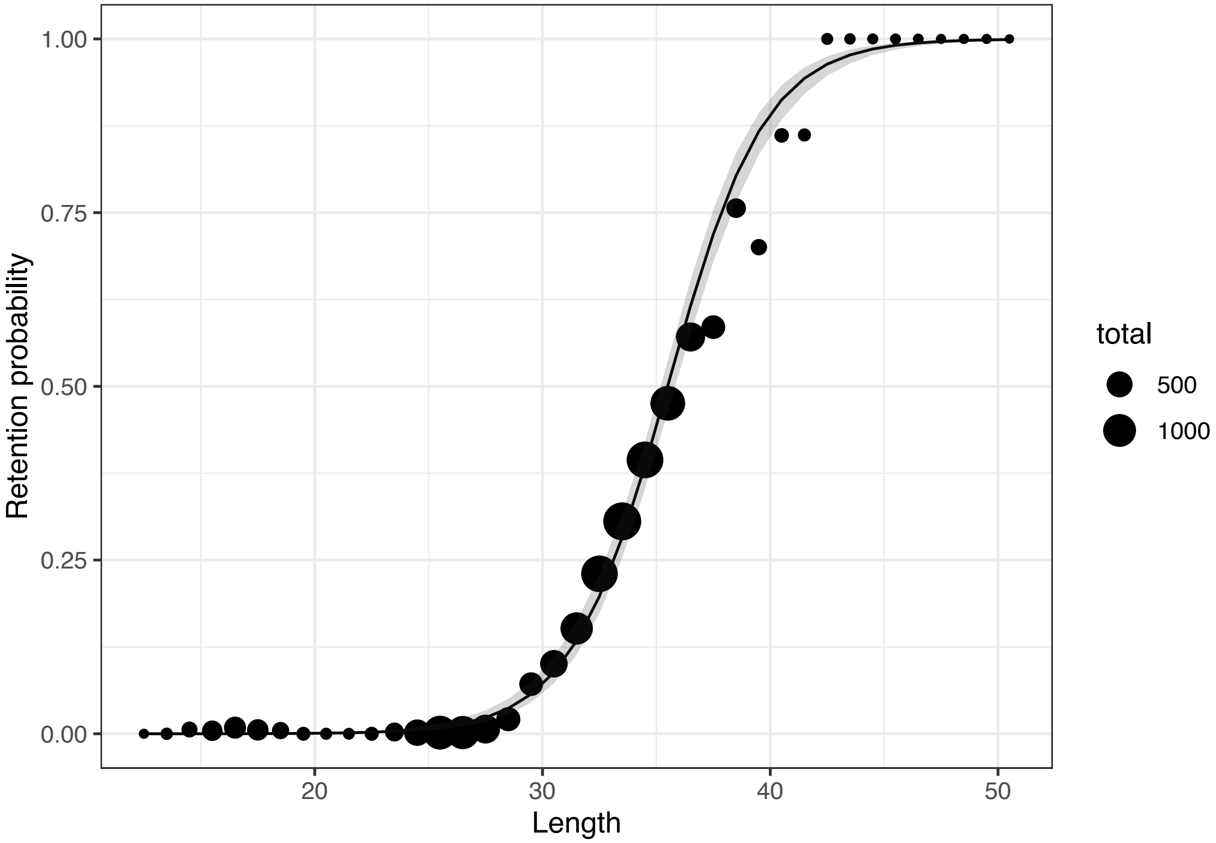
Windows bootstrapping in parallel

```
ncpus = 4
cl = makeCluster(rep("localhost", ncpus))
clusterExport(cl, "newdata")
bs = bootSel(mod_base, nsim=1000, parallel = "snow", cl=cl,
    FUN=function(mod){predict(mod, newdata=newdata, type="selection")})
stopCluster(cl)
```

Then we calculate quantiles across bootstraps for each row of `newdata`.

```
quants = apply(bs$t, 2, quantile, c(0.025, 0.5, 0.975))
newdata[,c("lo", "mid", "hi")] = t(quants)
```

```
ggplot(sumdat_base, aes(Length, prop))+geom_point(aes(size=total, y=raised_prop))+
   geom_line(data=newdata)+
   geom_ribbon(data=newdata, aes(ymin=lo, ymax=hi), alpha=0.2)+
    ylab("Retention probability")
```



All four codends in one model

We can analyze the data from all gear types together and test if the gear type affects selectivity by comparing models of varying complexity. If we want to use models directly (i.e. before bootstrapping) for testing the significance of a variable, we need to account for variability among hauls and avoid pseudoreplication (Hurlbert 1984) by including a random effect of haul. The random effect is only needed when models are used directly for hypothesis testing e.g. here, to test our hypothesis that the gear types differ in their selectivity.

```
mod0 = selfisher(prop~ Length +(1|haul), qratio=qratio, total=total, coverhaddock)
mod1 = selfisher(prop~ Length * gear +(1|haul), qratio=qratio, total=total, coverhaddock)

BICtab(mod0, mod1)
```

```
##      dBIC df
## mod1  0.0 9
## mod0 16.8 3
```

We use the Bayesian information criterion (BIC) to choose the best fit and this table tells us that `mod1` is more parsimonious, which means that the 4 codends are somehow different. To explore which aspect of the gears made them different, we examine models with mesh size, twine bending stiffness and catch size. Further, we assume that Baranov's Principle Of Geometric Similarity applies, i.e. selectivity is a function of fish length scaled by mesh size. It could also be reasonable to fit models with a main effect of mesh size and length in addition to an interaction term, but we do not attempt to fit all possible models here and leave it to users to decide what applies to their particular study.

```
mod_all = selfisher(prop ~ Length + (1|haul),
    qratio=qratio, total=total, haul=haul, data=coverhaddock)

mod_all_c = selfisher(prop ~ Length + catch + (1|haul),
    qratio=qratio, total=total, haul=haul, data=coverhaddock)

mod_all_s = selfisher(prop ~ Length + stiffness + (1|haul),
    qratio=qratio, total=total, haul=haul, data=coverhaddock)

mod_all_m = selfisher(prop ~ I(Length/mesh) + (1|haul),
    qratio=qratio, total=total, haul=haul, data=coverhaddock)

mod_all_c_s = selfisher(prop ~ Length + catch + stiffness + (1|haul),
    qratio=qratio, total=total, haul=haul, data=coverhaddock)

mod_all_c_m = selfisher(prop ~ I(Length/mesh)  + catch + (1|haul),
    qratio=qratio, total=total, haul=haul, data=coverhaddock)

mod_all_s_m = selfisher(prop ~ I(Length/mesh) + stiffness + (1|haul),
    qratio=qratio, total=total, haul=haul, data=coverhaddock)

mod_all_c_s_m = selfisher(prop ~ I(Length/mesh) + catch + stiffness + (1|haul),
    qratio=qratio, total=total, haul=haul, data=coverhaddock)

BICtab(mod_all, mod_all_c, mod_all_s, mod_all_m, mod_all_c_s, mod_all_c_m, mod_all_s_m,
  mod_all_c_s_m)
```

```
##               dBIC df
## mod_all_c_s_m  0.0 5
## mod_all_s      3.6 4
## mod_all_s_m    9.4 4
## mod_all_c_s   10.4 5
## mod_all       17.6 3
## mod_all_c     23.4 4
## mod_all_m     24.9 3
## mod_all_c_m   29.5 4
```

Again we use BIC to choose the best fit and this table tells us that the model that contains mesh size, twine bending stiffness and catch size is the most parsimonious fit to the data.

Bootstrapping

We drop the random effect of haul from the most parsimonious model before bootstrapping for two reasons (1) the bootstrapping method resamples among and within hauls and thereby accounts for variation among hauls, and (2) random effects make model fitting much slower which can be burdensome when refitting the model 1000 times or more.

```
mod_all_c_s_m_FE = selfisher(prop ~ I(Length/mesh) + catch + stiffness,
    qratio=qratio, total=total, haul=haul, data=coverhaddock)
```

Then we create a new data set to use for predictions. It must include all variables that appear in the model. Even though `haul` is not used in the mathematics behind the predictions, it must be included in the new data for technical reasons. We include `gear` just because it makes it easy to organize and plot the data further down. For each gear, we will make predictions over the range of catch weights observed for that gear.

```
newdata_v2 = unique(coverhaddock[,c("Length", "stiffness", "mesh", "gear", "catch")])

newdata_v2 = transform(newdata_v2,
    qratio=1,
    total=1,
    haul=NA)
```

Windows bootstrapping code

```
ncpus = 4
cl = makeCluster(rep("localhost", ncpus))
clusterExport(cl, "newdata")
bs = bootSel(mod_all_c_s_m_FE, nsim=1000, parallel = "snow", cl=cl,
    FUN=function(x){predict(x, newdata= newdata_v2, type="selection")})
stopCluster(cl)
```

Mac and linux bootstrapping code

```
bs= bootSel(mod_all_c_s_m_FE, nsim=1000, parallel = "multicore", ncpus = 4,
    FUN=function(x){predict(x, newdata= newdata_v2, type="selection")})
```

Organize bootstrap predictions with the predictor variables

Then we organize the bootstrap results and join them with the newdata used for predictions.

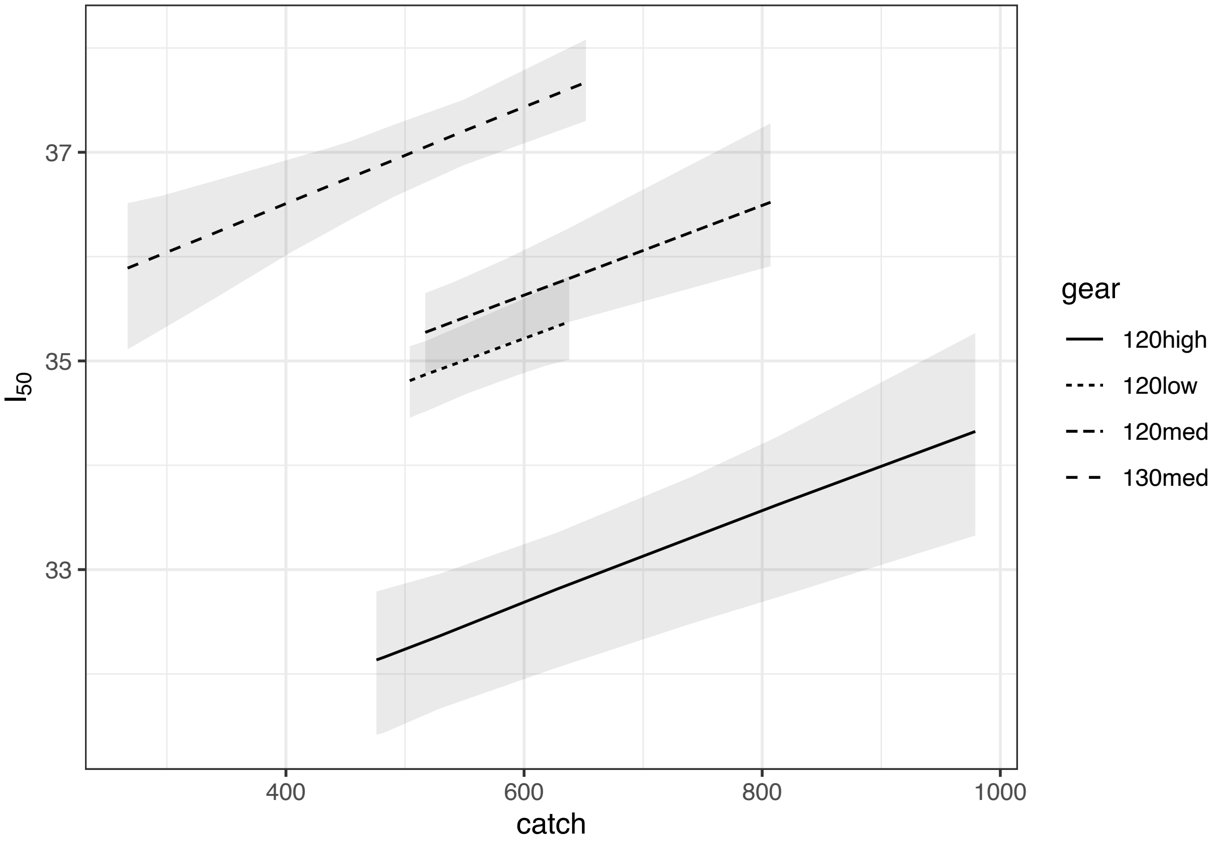
```
quants = apply(bs$t, 2, quantile, c(0.025, 0.5, 0.975))

newdata_v2[,c("lo", "mid", "hi")]=t(quants)

#all the bootstrap predictions with the variables used to create them
newdata_v3 = cbind(newdata_v2[,c("Length", "gear", "mesh", "stiffness", "catch")], t(bs$t))
```

```
#put them in long format (i.e. separate the different bootstrap replicates)
newdata_v3 = melt(newdata_v3, id.vars=1:5)

names(newdata_v3)[6:7] = c("rep", "predicted_r")
```

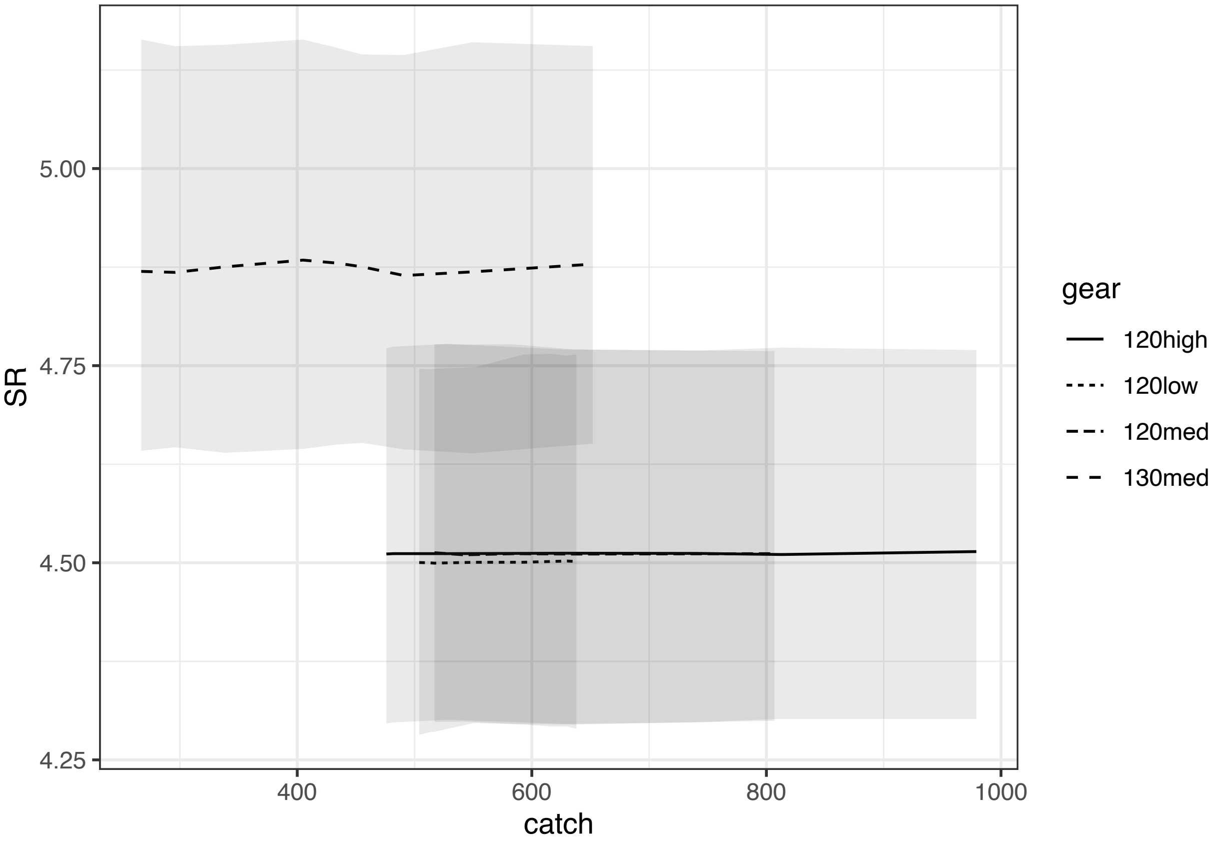
Calculate l_{50} and SR from bootstraps

Here we define a function to evaluate the length at which a given proportion of fish (p) are retained using interploation.We use this function to find the l_{50} and SR for each bootstrap while varying over mesh size, twine bending stiffness and catch size and subsequently find the mean l_{50}, mean SR and their 95% confidence limits. We plot the results against catch size, which is how they were presented in the original analysis of O'Neill et al (2016).

```
# Function to interpolate lengths - assumes y strictly increases with x
findx=function(x,y,p=0.5) {
  n=1:length(x)
  lo.obs=sum(y<p)
  hi.obs=lo.obs+1
  delta=(p-y[lo.obs])/(y[hi.obs]-y[lo.obs])
  x[lo.obs]+delta*(x[hi.obs]-x[lo.obs])
}

#find l50 and SR for each bootstrap, still varying by gear characteristics and catch
sums0 = ddply(newdata_v3, ~gear + mesh + stiffness + catch + rep,
    summarize, SR = findx(Length, predicted_r, 0.75) - findx(Length, predicted_r , 0.25),
    l50 =  findx(Length, predicted_r, 0.5)
)

#summarize over bootstraps, but still varying by gear characteristics and catch
sums1 = ddply(sums0, ~gear + mesh + stiffness + catch , summarize,
    l50_lo=quantile(l50, 0.025),
    l50_mid=quantile(l50, 0.5),
    l50_hi=quantile(l50, 0.975),
    SR_lo=quantile(SR, 0.025),
    SR_mid=quantile(SR, 0.5),
    SR_hi=quantile(SR, 0.975)
)

ggplot(sums1, aes(x=catch, group=gear))+
    geom_line(aes(y=l50_mid, lty=gear))+
    geom_ribbon(aes(ymin=l50_lo, ymax=l50_hi), alpha=0.1)+
    ylab(expression(l[50]))
```



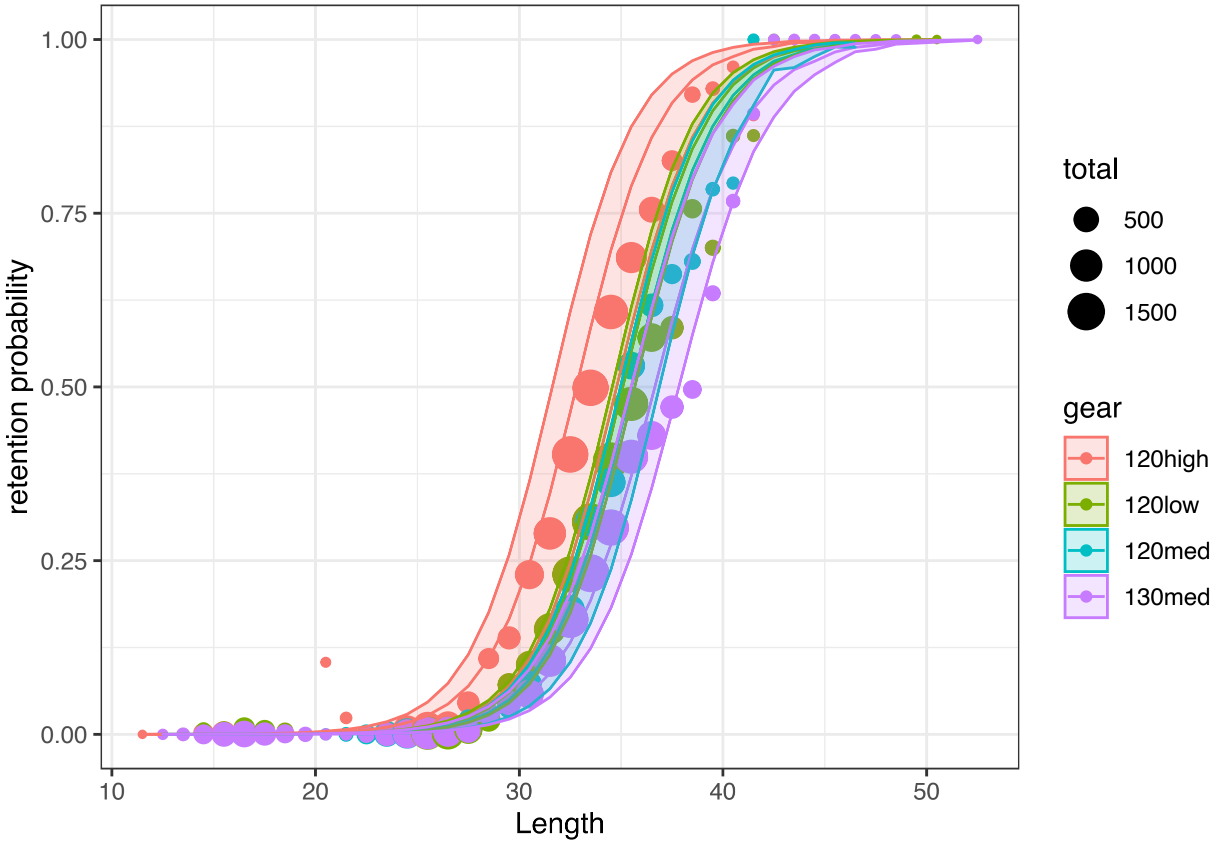
```r
ggplot(sums1, aes(x=catch, group=gear))+
    geom_line(aes(y=SR_mid, lty=gear))+
    geom_ribbon(aes(ymin=SR_lo, ymax=SR_hi), alpha=0.1)+
    ylab("SR")
```



The plots presented here are very comparable to Fig 1 from the original manuscript, which finds a similar dependence of haddock l_{50} on mesh size, twine bending stiffness and total codend catch weight but finds that SR is a constant (O'Neill et al., 2016). We should not, however, expect the results to be identical as there are many differences between the analyses. In the original, it was assumed that the slope and intercept of the logistic link functions vary randomly from haul to haul and that l_{50} and log SR mean selection curve were linearly dependent on the explanatory variables. Whereas, here we have assumed geometric similarity and that overall retention is related to the explanatory variables.

Below we plot retention curves for each gear aggregated over catches.

```
sumdat = ddply(coverhaddock, ~Length+gear, summarize,
    prop=sum(codend)/sum(total), total=sum(total),
    raised_prop=sum(codend * cod_rf)/sum(codend * cod_rf + cover* cov_rf),
    raised_total=sum(codend * cod_rf + cover* cov_rf)
)

newdata_v4 = ddply(newdata_v3,~Length + gear + mesh + stiffness , summarize,
    lo = quantile(predicted_r, 0.025),
    mid = quantile(predicted_r, 0.5),
    hi = quantile(predicted_r, 0.975)
)

ggplot(sumdat, aes(colour=gear))+geom_point(aes(size=total, x=Length, y=raised_prop))+
  geom_line(data=newdata_v4, aes(x=Length, y=mid))+
  geom_ribbon(data=newdata_v4, aes(x=Length, ymin=lo, ymax=hi, fill=gear), alpha=0.2)+
    ylab("retention probability")
```



References

Hartig F. (2020). DHARMa: Residual Diagnostics for Hierarchical (Multi-Level / Mixed) Regression Models. R package version 0.2.7. https://CRAN.R-project.org/package=DHARMa

Hurlbert S.H. (1984) Pseudoreplication and the design of ecological field experiments. Ecol. Monogr. 54:187–211.

O'Neill F.G., Kynoch R.J., Blackadder L., Fryer, R.J., Eryasar A.R., Notti, E., and Sala A., 2016. The influence of twine tenacity, thickness and bending stiffness on codend selectivity. Fisheries Research. 76, 94-99.

Millar R.B. (1994) Sampling from trawl gears used in sized selectivity experiments, ICES Journal of Marine Science, 51:293–298, https://doi.org/10.1006/jmsc.1994.1030

Appendix 2: Paired gear analysis of codend selectivity dependent on mesh size

12 Nov 2020

Case study background

This example deals with brown shrimp selectivity data from Santos et al. (2018). The brown shrimp beam-trawl fishery is one of the most important fisheries in the Southern North Sea. Despite its relevance, this is also one of the least regulated fisheries in European waters. Concerns regarding the size selection of the fishery motivated the German research project *CRANNET* (2013-2015), which assessed brown shrimp size selection of commercially used and alternative codend designs. Codend selectivity data was collected during experimental fishing trials using the paired gear method (Millar and Walsh, 1992; Wileman, 1996). The experimental method consisted of fishing with two identical beam trawls, simultaneously and in parallel on the same shrimp population. One of the trawls mounted a small-mesh (11 mm) control codend with very limited selectivity (assumed to be non-selective) on the range of shrimp lengths available for the trawl. The second trawl mounted, one at a time, a total of 33 different test codend designs varying in mesh size and mesh type were tested.

This case study draws on a subset of the *CRANNET* data to demonstrate the use and functionality of `selfisher` in selectivity analysis based on paired gear data. It uses a subset of the catch data, the 87 hauls during which 13 diamond-mesh codends varying in mesh size (ranging from 19.1 mm to 36.3 mm) were tested. Additional information relative to fishing conditions and catch characteristics were recorded at haul level.

Preliminaries

```
library(selfisher)
library(xtable)
library(bbmle)
library(ggplot2); theme_set(theme_bw())
```

Data structure

We can read in the data that is distributed with the package and examine it.

```
data(pairedshrimp)
head(pairedshrimp)
```

```
##   length     ms     cw seast test control sampling_test sampling_control haul
## 1     23 20.188 12.074     4    0       1         0.232            0.182    1
## 2     24 20.188 12.074     4    1       1         0.232            0.182    1
## 3     26 20.188 12.074     4    0         1          0.232             0.182     1
## 4     28 20.188 12.074     4    0         2          0.232             0.182     1
## 5     29 20.188 12.074     4    0         3          0.232             0.182     1
## 6     30 20.188 12.074     4    2         1          0.232             0.182     1
```

Here we can see the column names of the data. There is a column for haul because multiple hauls $h, \ i = 1, 2, \ldots, 87$ are contained in the same data object. Therefore the information presented in the remaining columns was collected at haul level. The column length contains observed (total) length classes l of brown shrimp (mm), ms contains the measured mesh size (mm) of the test codend, cw is the total catch weight (kg) collected in the test codend, seast is the state of the sea recorded during towing (Beaufort scale). Columns test and control contain the numbers of shrimps of length l sampled from each of the paired codends, while sampling_test and sampling_control are the associated sampling fractions.

Data transformation

In selfisher, it is necessary to model the binomial response as a proportion and total because unlike many other methods for binomial GLMs, the underlying code does not accept a two column response variable. Therefore, we transform the data to create these new columns. We also create a column for qratio (i.e. q_h below) which is the ratio of the sampling fraction of the test over the control gear. If the data contained raising factors instead, then qratio would be the raising factor of the control gear over the test gear.

```
pairedshrimp = transform(pairedshrimp, total = test + control,
                 qratio = sampling_test / sampling_control,
                 prop = test / ( test + control )
)
```

Selectivity analysis

The selectivity analysis based on paired gear data is usually done with the model introduced by Millar and Walsh (1992):

$$\phi(l) = \frac{p * r(l)}{(1.0 - p) + p * r(l)}$$

In the equation above, $\phi(l)$ expresses the probability that a shrimp of length l in the total catch of the paired gear was caught by the test gear. This probability is related to a sequence of two fishing events. The first event is controlled by the probability that a shrimp entering the paired gear did it through the test trawl (relative fishing power), expressed as the length-independent split parameter p. Conditioned on the probability of entering the test trawl, the second event is the length-dependent retention probability of the test codend $r(l)$. Retention probability is usually (but not exclusively) modelled with the logistic function, and summarized by two selectivity parameters L_{50} (length of 50% retention probability) and SR (difference in the lengths of 75% and 25% retention). Further details of this model and extensions involving alternatives to the logistic function can be found in Millar and Walsh (1992) and Wileman (1996).

In Santos *et al.* (2018), the effect of codend design as well as other variables describing catch and operational characteristics of the hauls were assessed using the so-called Fryer method (Fryer, 1991). The Fryer method is carried out in two steps. In the first step the parameters l_{50}, SR, p and associated Hessian-based covariance-variance matrix of individual hauls are estimated. The estimates become the data used in the second step, where the effect of the measured explanatory variables (fixed effects) on codend selectivty is estimatated. Such estimations are obtained using the EM-algorithm, which allows quantifying the strength of the fixed effects in the presence of between-haul variation.

The Fryer method was developed at a time before suitable generalized mixed modeling software such as `selfisher` was available. The Fryer method suffers from small sample bias in the fits to individual haul data. In contrast, `selfisher` enables quantifying and testing the effect of explanatory variables in a single step and directly on the measured catch data.

The methods in `selfisher` generalize the original selectivity model introduced by Millar and Walsh (1992), by allowing multiple fixed and random effects to simultaneously model relative fishing efficiency and the selectivity of the test codend:

$$\phi_{h,l} = \frac{q_h p_h r_{h,l}}{(1.0 - p_h) + q_h p_h r_{h,l}}$$

In the equation above, $q_h = \frac{s_{t,h}}{s_{c,h}}$ is the ratio of the fraction of brown shrimp sampled in the test gear to the fraction sampled in the control gear, p_h is the relative fishing power of the test gear in haul h (split parameter), where $p_h = logit^{-1}(\mu + u_h)$ and u_h is a random effect $u_h \sim N(0, \sigma_u^2)$ accounting for haul-specific random variation from the mean value on the logit scale μ. For simplicity in this example, we assume a logit link in the retention model and therefore the retention probability model for haul h and length l is $r_{h,l} = logit^{-1}(y_{h,l}) = \frac{exp(y_{h,l})}{1+exp(y_{h,l})}$). This expresses the haul-specific probability for a fish entering in the test gear to be retained where $y_{h,l}$ is the expectation on the link scale combining fixed- and random-effects potentially influencing retention probability of the test codend. Four models varying in the structure of $y_{h,l}$ are initially considered:

$$\begin{aligned}
model\ 1: &\quad y_{h,l} = \beta_0 + a_h + (\beta_1 + \tau_h) * l \\
model\ 2: &\quad y_{h,l} = \beta_0 + a_h + (\beta_1 + \tau_h) * l + \beta_2 * ms_h \\
model\ 3: &\quad y_{h,l} = \beta_0 + a_h + (\beta_1 + \tau_h) * l + \beta_2 * ms_h + \beta_3 * l * ms_h \\
model\ 4: &\quad y_{h,l} = \beta_0 + a_h + (\beta_1 + \tau_h) * l / ms_h
\end{aligned}$$

All models listed above include an intercept β_0 which expresses the baseline shrimp retention log-odds when all additional covariates (including shrimp length) are set to 0. The coefficient a_h is a random effect $a_h \sim N(0, \sigma_a^2)$ accounting for haul-specific random variation of the intercept. Model 1 assumes that codend retention can be exclusively described by shrimp length. In models 1, 2, and 3, β_1 is the slope quantifying the rate of increment in retention probability (on the link scale) due to increments in the length of shrimp. The slope of size selection curves can vary significantly and randomly between hauls, even while keeping the design characteristics of the codend constant (Fryer, 1991). The coefficient $\tau_h \sim N(0, \sigma_\tau^2)$ is introduced to account for haul-specific random variation of the baseline β_1 value. Model 2 extends model 1 by adding mesh size (ms) as a fixed effect that varies by haul: ms_h. Model 2 assumes that retention probability is a result of separate effects of shrimp length and mesh size. With the addition of an interaction term of mesh size and shrimp length (β_3), in model 3 it is assumed that mesh size influences both the position and slope of the size selection curve.

The tested diamond-mesh codends were made of the same netting material, same length, and the number of meshes in circumference were reduced proportionally to increments in mesh size. Therefore, we assume that the meshes of the tested codends present the same geometry during fishing. Based on the principle of geometrical similarity (Baranov, 1948; Millar and Holst, 1997; Revill and Holst, 2004) it would be a reasonable *a priori* assumption that the selection curves will vary systematically with mesh size. Model 4 is built upon the hypothesis that variation in size selection can be explained by the principle of geometrical similarity, implying that the size selection varies proportionally to mesh size (i.e. doubling the mesh size implies doubling the values of l_{50} and SR of the selection curve)(Millar and Holst, 1997).

Model fitting with `selfisher`

In this section, the models described above are fit using a function named `selfisher` within the `selfisher` package. Formulas in these models follow the convention of the `lme4` and `glmmTMB` packages. The function takes a formula for the retention model `rformula` as the first argument. This is a two sided model with the proportion on the left side and fixed and random effects on the right side. For example, in model 1 (`m1`) below, `prop ~ length + ( length | haul )` says that the proportion (`prop`) of fish retained in the test gear depends on `length` and that the intercept and slope vary randomly by haul. It also takes an argument for the relative fishing power model (`pformula`) which is a one-sided formula, e.g. `pformula =~( 1 | haul )`; this says that relative fishing power should be estimated and can vary randomly by haul. If instead, we wanted to fix the split at 0.5, then we could specify `pformula =~0`. To tell the funciton that this is a paired gear model with one non-selective gear, we use the argument `psplit = TRUE`. The function also takes as arguments the names of the columns for the `total` and `qratio` within the data frame specified by the argument `data`.

One optional argument appears in this example: the `start` argument. It tells the function what starting values to use in maximum likelihood estimation. See `?selfisher` for the full flexibility of how to specify starting values, but here we only give starting values for the retention model. To get good starting values for the retention model's intercept (β_0 above) and coefficient on `length` (β_1 in models 1, 2, and 3 above), we use the `inits` function which takes guesses for l_{50} and SR as its arguments (30 and 8 respectively here). The coefficient β_1 has a different meaning in model 4, but the starting values work as supported by plots below, so it's not a problem. Models fit with `TMB` (as in `selfisher`) are usually robust to starting values, but due to the complexity of paired gear models, they sometimes get stuck in local minima during maximum likelihood estimation. In this case study, the length of the `start` vector must equal the number of fixed effects coefficients in the retention model, i.e. the βs in the equations above, so in `m2` and `m3` we combine guesses for β_0 and β_1 with zeros for the other coefficients

As described above, model 4 (`m4`) assumes geometric similarity. To include $length/ms$ in a model in `R`, it is necessary to tell the formula interface to use the term "as is" by putting an $I()$ around it.

```
m1 = selfisher( prop ~ length + ( length | haul ),
                total = total, psplit = TRUE, pformula =~( 1 | haul ),
                qratio = qratio, data = pairedshrimp, start = c(inits(30,8))
                )

m2 = selfisher( prop ~ length  +  ms  + ( length | haul ),
                total = total, psplit = TRUE, pformula =~( 1 | haul ),
                qratio = qratio, data = pairedshrimp, start = c(inits(30,8),0)
                )

m3 = selfisher( prop ~ length * ms  + ( length | haul ),
                total = total, psplit = TRUE, pformula =~( 1 | haul ),
                qratio = qratio, data = pairedshrimp, start = c(inits(30,8),0,0)
                )

m4  = selfisher( prop ~  I( length/ms ) + ( I( length/ms ) | haul ),
                total = total, psplit = TRUE, pformula =~( 1 | haul ),
                qratio = qratio, data = pairedshrimp, start = c(inits(30,8))
                )
```

Model selection

Retention is a mechanical process that can be largely explained by the relationship between the mesh characteristics and the morphology of the species being selected. Of the above models, model 4 is the only one that has a mechanistic justification, namely geometric similarity. In terms of model selection we *a priori* propose model 4, taking the view that strong evidence against model 4 is required to prefer an alternative. There are several thousand observations in the data, and consequently Akaike Information Criterion will tend to choose the most complicated model (Heinze *et al.*, 2018). In contrast, the Bayesian Information Criterion, BIC, includes a stronger penalty for the number of parameters than AIC and therefore it tends to select simpler models than AIC. We can use functions from the bbmle package to create either an AIC or BIC table of the models.

```
AICtab(m1, m2, m3, m4)
```

```
##    dAIC df
## m4  0.0 7
## m3  0.2 9
## m2  7.5 8
## m1 96.6 7
```

```
BICtab(m1, m2, m3, m4)
```

```
##    dBIC df
## m4  0.0 7
## m3 13.1 9
## m2 13.9 8
## m1 96.6 7
```

Both AIC and BIC rank model 4 on top, as was the *a priori* expectation.

Extended models

Model 4 is now extended by adding other covariates in the data frame that could potentially influence size selection, such as cw and seast. The following models are fitted following the same procedure as for models 1 to 4:

```
m5 = selfisher( prop ~  I( length/ms )+ cw + ( I( length/ms ) | haul ),
                total = total, psplit = TRUE, pformula =~( 1 | haul ),
                qratio= qratio, data = pairedshrimp, start = c(inits(30,8),0)
                )

m6 = selfisher( prop ~ I( length/ms ) + seast + ( I( length/ms ) | haul ),
                total = total, psplit = TRUE, pformula =~( 1 | haul ),
                qratio= qratio, data = pairedshrimp, start = c(inits(30,8),0)
                )

m7 = selfisher( prop ~ I( length/ms ) + cw + seast + ( I( length/ms ) | haul ),
                total = total, psplit = TRUE, pformula =~( 1 | haul ),
                qratio = qratio, data = pairedshrimp, start = c(inits(30,8),0,0)
                )

m8 = selfisher( prop ~ I( length/ms ) + cw * seast +( I( length/ms ) | haul ),
                total = total, psplit=TRUE, pformula =~( 1 | haul ),
                qratio = qratio, data = pairedshrimp, start = c(inits(30,8),0,0,0)
                )
```

Then we can compare all the models.

```
BICtab(m1, m2, m3, m4, m5, m6, m7, m8)
```

```
##    dBIC df
## m4  0.0 7
## m5  2.4 8
## m6  8.3 8
## m7  8.6 9
## m8 13.0 10
## m3 13.1 9
## m2 13.9 8
## m1 96.6 7
```

This table shows that none of the individual or combined effects associated with `cw` and `seast` were strong enough to be included in the most parsimonious model. Therefore model 4 is selected in this case study for further assessments.

Simple results

Consistent with other model procedures implemented in `R`, a summary of model results and fit statistics can be obtained via `summary(model.object)`. Before inspecting the size selectivity results provided by model 4, the estimated relative fishing efficiency of the test gear is presented. By default, `selfisher()` summary shows the estimated split parameter p on the logit scale, but this might be updated in a new version. An inverse logit transformation is needed to obtain the fishing power $p \in [0, 1]$:

```
round(boot::inv.logit(confint(m4, level=0.95, component="p")[1,c( 1, 3, 2 )]),3)
```

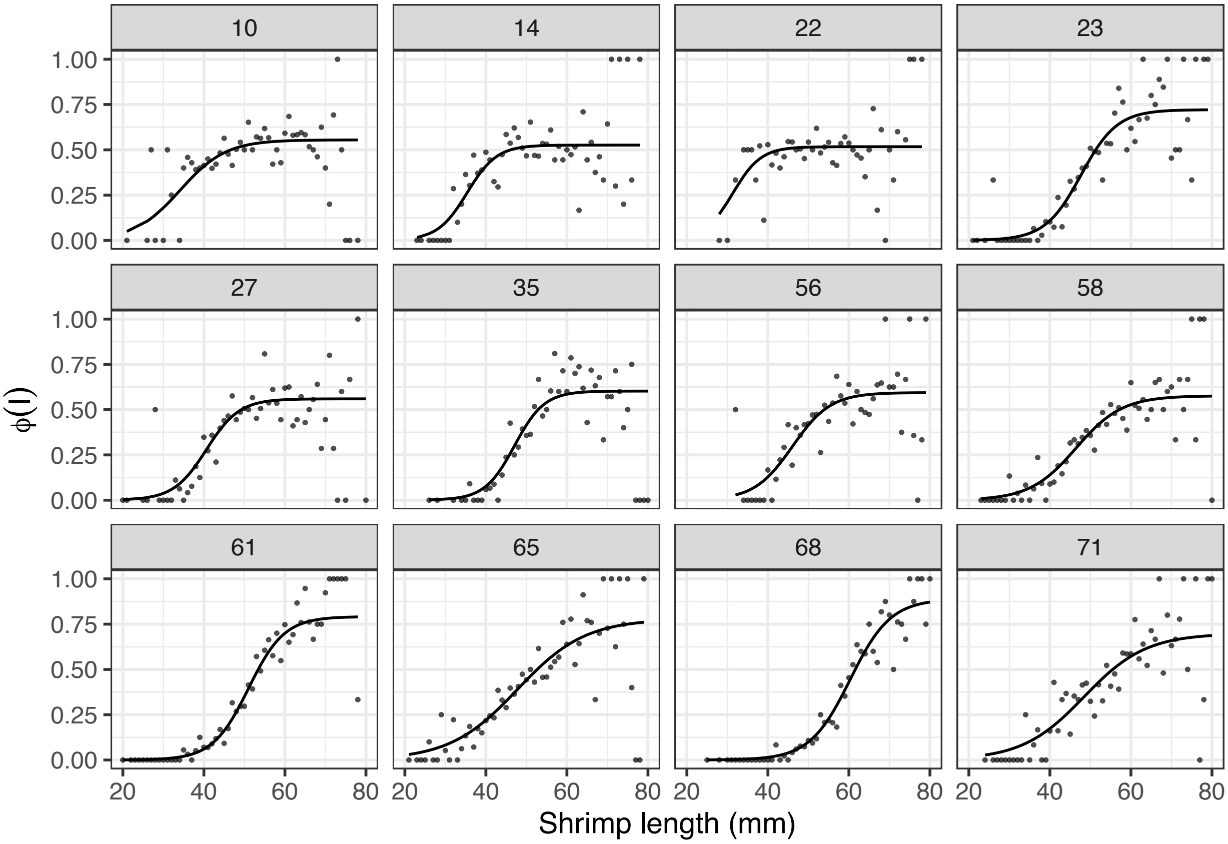
```
##    2.5 % Estimate   97.5 %
##    0.484    0.503    0.522
```

The split parameter estimated by model 4 is $p = 0.503$ with 95% confidence interval $(0.484 - 0.522)$, very close to the value estimated in the original study ($p = 0.492$ $(0.472 - 0.512)$). This includes the reference value of 0.5, from which we conclude that there is no significant evidence against equal catch efficiency of the test and control gears.

Goodness-of-fit

The mixed models presented above are fitted at haul level. Therefore it is reasonable to assess the goodness-of-fit of these models on individual haul data. Due to the large dataset used, a random sample of 12 hauls are picked to demonstrate how well the $\phi(l)$ curves estimated by model 4 fit to the data. We can use the function `predict` to examine the estimated retention curve, ie. the model's "response" variable. See `?predict.selfisher` for details of how to use this function including the different types of predictions available.

```
pairedshrimp$hat_phi = predict( m4, type = "response" )

set.seed( 999 )

ggplot(pairedshrimp[pairedshrimp$haul%in%sample(unique(pairedshrimp$haul),12),],
          aes(x=length,y=prop))+
          ylab( eval( parse( text = paste0("expression(","phi", "(l))") ) ) )+
          xlab( "Shrimp length (mm)" )+
          xlim( 20, 80 )+
          geom_point( alpha = .7, size = 0.3 )+
          geom_line( aes( x = length, y = hat_phi ) )+
          facet_wrap( ~ haul, ncol = 4, nrow = 3)+
          theme_bw()
```



The figure above reveals that the fitted $\phi(l)$ curves describe well the trends and variability of the data at individual haul level.

Population-average predictions

Mixed modelling is a formal procedure that takes into account specific details of the data collection enabling sound hypothesis testing on fixed effects and model selection. However, fitting mixed models can be a computational-intensive task. Moreover, the researcher is also typically interested in obtaining average selectivity predictions, as these are relevant to the selectivity applied to the fishery. It is therefore recommended to refit the best candidate model leaving out the random effects. Bootstrapping can then be used to obtain valid standard errors and confidence intervals for estimated quantities such as l_{50} and SR.

```
m4_fe = selfisher( prop ~ I( length/ms ), total = total,
                  psplit = TRUE, pformula =~1, haul = haul,
                  qratio = qratio, data = pairedshrimp, start = c( inits( 30, 8 ) ) )
```

A summary of model coefficients describing codend retention can be obtained using standard procedures in R

```
coef( summary( m4_fe ) )$r
```

```
##                Estimate Std. Error   z value Pr(>|z|)
## (Intercept)   -9.231103 0.09420144 -97.99322        0
## I(length/ms)   5.143515 0.06443302  79.82732        0
```

Bootstrapping to get confidence intervals on population-average l_{50} and SR

Selectivity statistics l_{50} and SR are often obtained by simple calculations involving model coefficients β_0 and β_1, or for simple models in `selfisher`, the function `L50SR(model.object)` can calculate them. However, standard calculations need to be updated when using multiple fixed effects to describe codend retention. Details on how to calculate l_{50} and SR from the models considered in this case study can be found in Table 1 below.

To obtain a bootstrap distribution of the selectivity parameters estimated by model 4, first we generate a bootstrap distribution of model coefficients using `bootSel()`, as follows. The `bootSel()` function applies the user-defined function `FUN` to each refit model; here we define `FUN` so that it returns the fixed effect (`fixef`) coefficients of the retention model (`$r`). This is the code to perform the computations in parallel on Linux or Mac computers, but see the other case studies for how to do it in Windows.

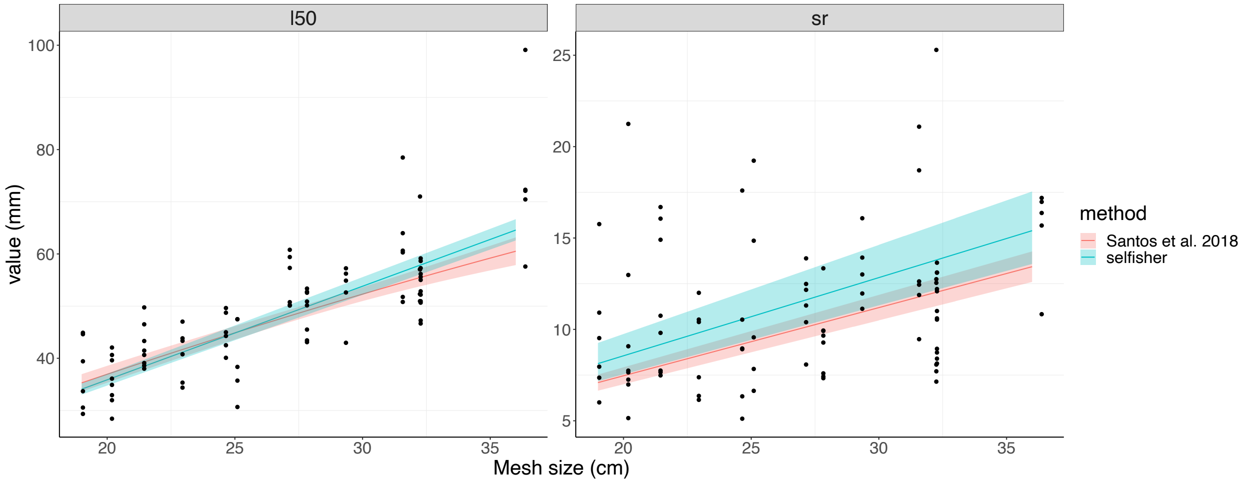
```
bootpars_m4_fe = bootSel( m4_fe, nsim = 1000,
                  parallel = "multicore", ncpus = 4,
                  FUN = function( x ){fixef( x )$r }
)
```

Selectivity statistics l_{50} and SR are then calculated from each set of coefficients in the bootstrap distribution generated above. Finally, the resulting bootstrap distribution is used to obtain 95% percentile confidence intervals of l_{50} and SR.

```
# extract boostrap distribution from bootSel object
bootpars = bootpars_m4_fe$t

# create a grid of mesh sizes within the experimental range for predictions
ms = seq( from = 20, to = 35, by = .5 )

# calculate bootstrap distribution of l50 based on  model coefficients
##  l50 =  -m * beta_0 / beta_1 (see table 3)
L50_boot = apply( bootpars, 1, function( x ){ -ms * x[1] / x[2] } )

# get Efron confidence intervals for l50
L50_ci = t( apply( L50_boot, 1, quantile, c ( 0.025, 0.5, 0.975 ) ) )

# create a data frame for plotting
L50_df = data.frame( mesh_size = ms )

L50_df [, c("lower_limit", "median", "upper_limit" )] = L50_ci

# calculate bootstrap distribution of SR based on  model coefficients
## SR =  m*log(9) / beta_1 (see table 3)
SR_boot = apply( bootpars, 1, function( x ){ ms * log(9) / x[2] } )

# get Efron confidence intervals for SR
SR_ci = t( apply( SR_boot, 1, quantile, c( 0.025, 0.5, 0.975 ) ) )

# create a data frame for plotting SR
```

```r
SR_df = data.frame( mesh_size = ms )

SR_df[, c( "lower_limit", "median", "upper_limit" )] = SR_ci
```

The following figure compares the predictions of l_{50} and SR estimated by model 4, with those from Santos et al. (2018). The black points in the figure represent values of l_{50} and SR estimated at haul level, and used in the original study as input data. Average predictions for l_{50} and estimation uncertainty (expressed in the amplitude of the confidence band) by model 4 are equivalent to those from the original study. Model 4 predicted higher values of SR , with larger uncertainty than Santos et al. (2018). This is a plausible result considering the different model structures applied, and the large variation of the by-haul estimates. Moreover, there is not statistical evidence to reject the possibility that the true value of SR could fall within the continue region of confidence bands overlap.



model	l50	SR
1	$\frac{-\alpha}{\beta_1}$	$\frac{log(9)}{\beta_1}$
2	$\frac{-(\alpha+\beta_2 * ms)}{\beta_1}$	$\frac{log(9)}{\beta_1}$
3	$\frac{-(\alpha+\beta_2 * ms)}{\beta_1+\beta_3 * ms}$	$\frac{log(9)}{\beta_1+\beta_3 * ms}$
4	$\frac{-\alpha * ms}{\beta_1}$	$\frac{log(9) * ms}{\beta_1}$
5-8	$\frac{-(\alpha+\sum_{i=2}^{p}\beta_i * x_i) * ms}{\beta_1}$	$\frac{log(9) * ms}{\beta_1}$

Table 1: Calculations to obtain selectivity parameters from population-average models 1-8 (fitted leaving out the random effects). Note that the last row for models 5-8 is the extension of model 4 to include additional covariates $x_i, i = 2, ..., p$.

References

Baranov, F. 1948. Theory and assessment of fishing gear. Pishchepromisdat, Moscow.

Fryer, R. 1991. A model of between-haul variation in selectivity. ICES Journal of Marine Science, 48: 281–290. Oxford University Press.

Heinze, G., Wallisch, C., and Dunkler, D. 2018. Variable selection–a review and recommendations for the practicing statistician. Biometrical Journal, 60: 431–449. Wiley Online Library.

Millar, R., and Walsh, S. 1992. Analysis of trawl selectivity studies with an application to trouser trawls. Fisheries Research, 13: 205–220. Elsevier.

Millar, R. B., and Holst, R. 1997. Estimation of gillnet and hook selectivity using log-linear models. ICES Journal of Marine Science, 54: 471–477. Oxford University Press.

Revill, A. S., and Holst, R. 2004. Reducing discards of north sea brown shrimp (c. Crangon) by trawl modification. Fisheries Research, 68: 113–122. Elsevier.

Santos, J., Herrmann, B., Stepputtis, D., Günther, C., Limmer, B., Mieske, B., and Schultz, S. *et al.* 2018. Predictive framework for codend size selection of brown shrimp (crangon crangon) in the north sea beam-trawl fishery. PloS one, 13: e0200464. Public Library of Science San Francisco, CA USA.

Wileman, D. A. 1996. Manual of methods of measuring the selectivity of towed fishing gears. ICES Coop. Res. Rep., 215: 1–126.

Appendix 3: Catch comparison on unpaired hauls

07 Dec 2020

This example deals with data from an experiment originally published by Savina et al. (2017). Two soak tactics, i.e., 12h at day and 12h at night, were compared in the Danish gillnet plaice fishery to estimate whether a change in soak tactics could help to catch less of the unwanted bycatch, i.e., the invertebrate edible crab (*Cancer pagurus*). The method developed by Herrmann et al. (2017) for assessing the relative length-dependent catch efficiency effect of changing from soak tactic *Day* to *Night* was used. This example is representative of experimental fishing where the catch data obtained for two different gear designs were not collected in pairs, and can allow for a different number of deployments.

Preliminaries

```
library(selfisher)
library(plyr) # for aggregating data across hauls
library(ggplot2); theme_set(theme_bw())
library(splines) # for bs function
```

Data structure

Load the data and check out the variables. This is a subset of the original dataset (one species, two soak durations). The data contains the length measurement of each individual to the nearest mm below (carapace width), as specified in the column "width". Every day for 7 days (I to VII), three fleets (each consisting of three gillnets tied together, and labelled A, B and C) were soaked for 12 h during the day (*Day*) and three others during the night (*Night*). Each deployment of a fleet is considered as a "haul" (with haul name written as Day_Soak_Fleet). Gear unit design is the soak tactic, specified in the column "tactic", with two levels: 12h at day (*Day*) and 12h at night (*Night*). "total" gives the number of individuals for each length class and haul. There was no sub-sampling.

```
data("compcrab")
head(compcrab)
```

```
## # A tibble: 6 x 4
##   width haul        tactic total
##   <dbl> <fct>       <fct>  <int>
## 1  58.5 VII_Night_B Night      1
## 2  63.5 VI_Night_B  Night      1
## 3  64.5 I_Night_B   Night      1
## 4  66.5 VII_Day_A   Day        1
## 5  69.5 VII_Night_B Night      1
## 6  70.5 VI_Night_B  Night      1
```

```
summary(compcrab)
```

```
##      width                 haul       tactic        total
##  Min.   : 58.5   VII_Night_A: 49   Day  :117   Min.   :1.000
##  1st Qu.: 99.5   VI_Night_B : 38   Night:446   1st Qu.:1.000
##  Median :115.5   V_Night_B  : 34               Median :1.000
##  Mean   :117.2   VII_Night_B: 33              Mean   :1.181
##  3rd Qu.:133.5   V_Night_A  : 28              3rd Qu.:1.000
##  Max.   :197.5   I_Night_B  : 27              Max.   :5.000
##                  (Other)    :354
```

Here we can see that all hauls are contained in one data frame, organized into what is called "long format", with *Day* and *Night* one after the other (unpaired).

Transforming data

For a model in `selfisher`, we need to convert counts into proportions and totals. We use the 'ddply' function to calculate the proportion of fish entering one of the gear design (here *Night*) for each length class and haul, i.e., 1 for *Night* and 0 for *Day*.

```
dat = ddply(compcrab, ~width+haul+tactic, mutate,
                prop = as.numeric(tactic == "Night")
                )
```

Catch comparison

The following is a typical model for catch comparison of multiple haul data without subsampling using spline with the `bs` function.

```
mod = selfisher(prop ~ bs(width, df = 3), total = total, haul = haul, pool = tactic, data = dat)
```

This models the proportion of fish in *Night* versus *Day* (`prop`) as a function of `width`. The `selfisher` function takes the `total` number of fish in *Day* and *Night* using a separate argument, `total`. The argument `haul` needs to be specified in order to perform double-bootstrapping as demonstrated below. Otherwise, it could be omited from the model specification as it doesn't affect the fit. The `haul` argument tells the software how to group the data for resampling in the bootstrapping procedure. `pool` represents the different pools of hauls, i.e., one for each soak tactic, that is used in double bootstrap to produce same number of hauls by pool. Indeed, because the catch data obtained for *Day* and *Night* were not collected in pairs (and may not have the same total number of deployments), we sum data of the deployments carried out with *Day*, and data of the deployments carried out with *Night*.

Then we create a new data set to make predictions on.

```
newdata = expand.grid(width = unique(dat$width),
                total = 1,
                haul = 0,
                tactic = NA) #not used as a predictor, so it doesn't matter

newdata$prop = predict(mod, newdata = newdata, type = "response")
```

Bootstrap to get CI on predictions

The code below runs in parallel on Mac and Linux computers as written here, but a Windows version was given above. This call to the function `bootSel` predicts the response variable based on the model `mod` and the covariates in `newdata`. Then we calculate the quantiles of the bootstrapped response variable, and transform the proportion into a catch ratio.

```
bs = bootSel(mod, nsim = 100, parallel = "multicore", ncpus = 4,
                     FUN = function(mod){predict(mod, newdata = newdata, type = "response")})

quants = apply(bs$t, 2, quantile, c(0.025, 0.5, 0.975))
newdata[,c("lo", "mid", "hi")] = t(quants)
```

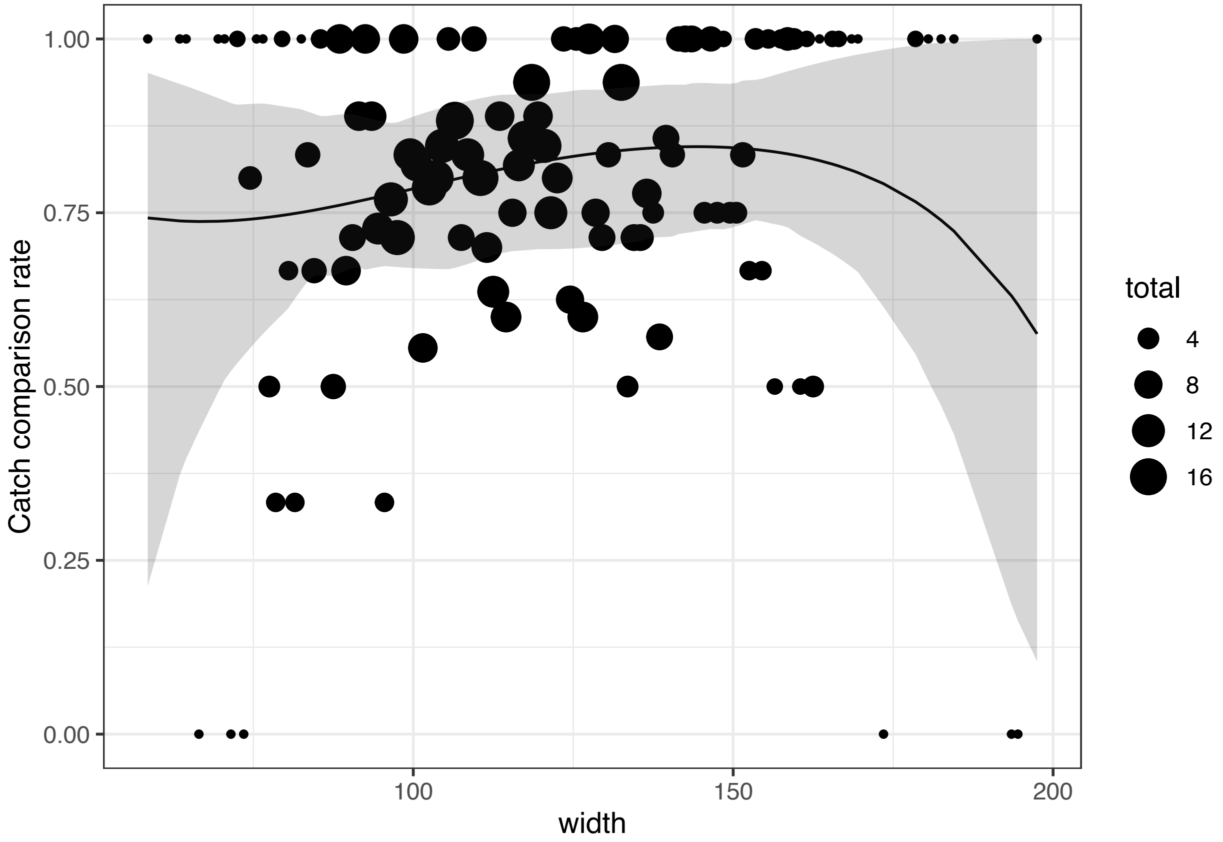
```
bs$CR = bs$t/(1-bs$t)
CRquants = apply(bs$CR, 2, quantile, c(0.025, 0.5, 0.975))
newdata[,c("CRlo", "CRmid", "CRhi")] = t(CRquants)
```

Plot predictions

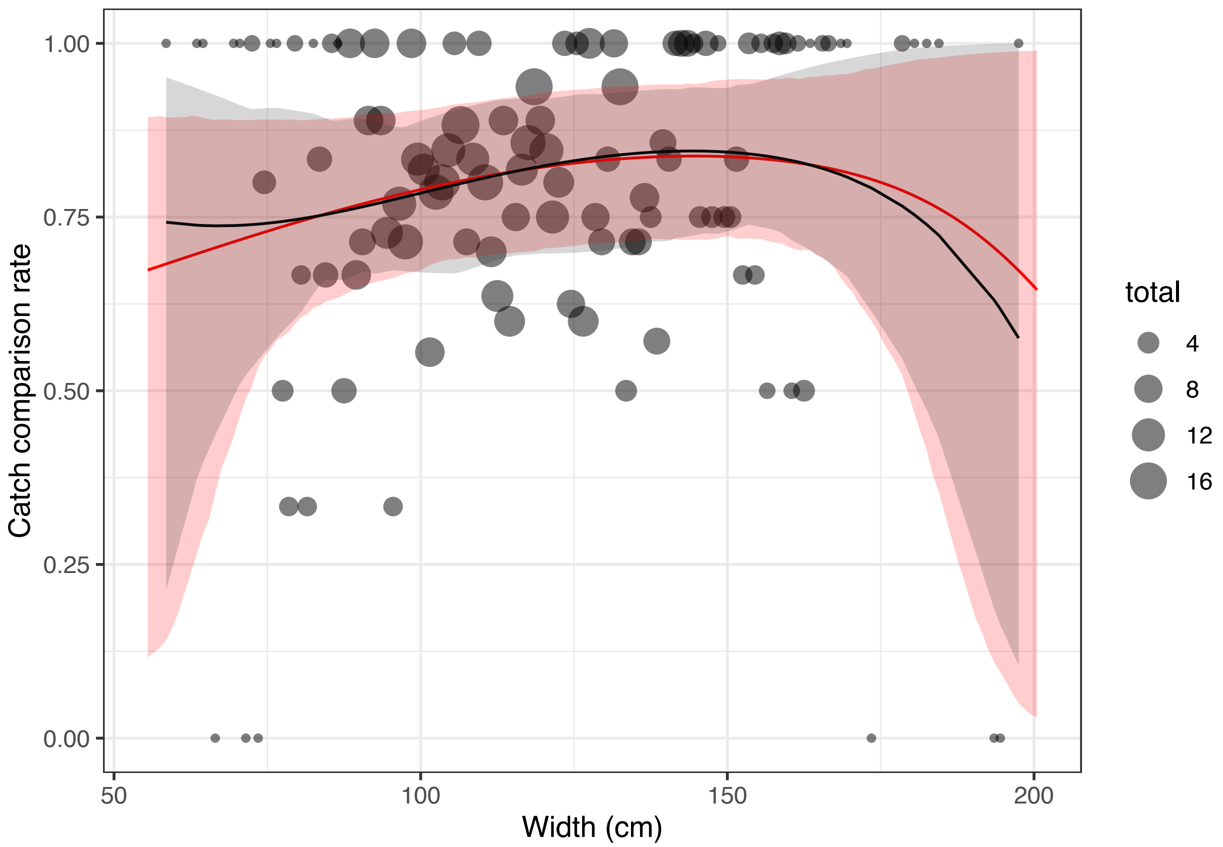
For plotting, we need to aggregate the hauls.

```
sumdat = ddply(dat, ~width, summarize,
                prop = sum(total*prop)/sum(total),
                total = sum(total),
                CR = sum(prop)/sum(1-prop)
                )
```

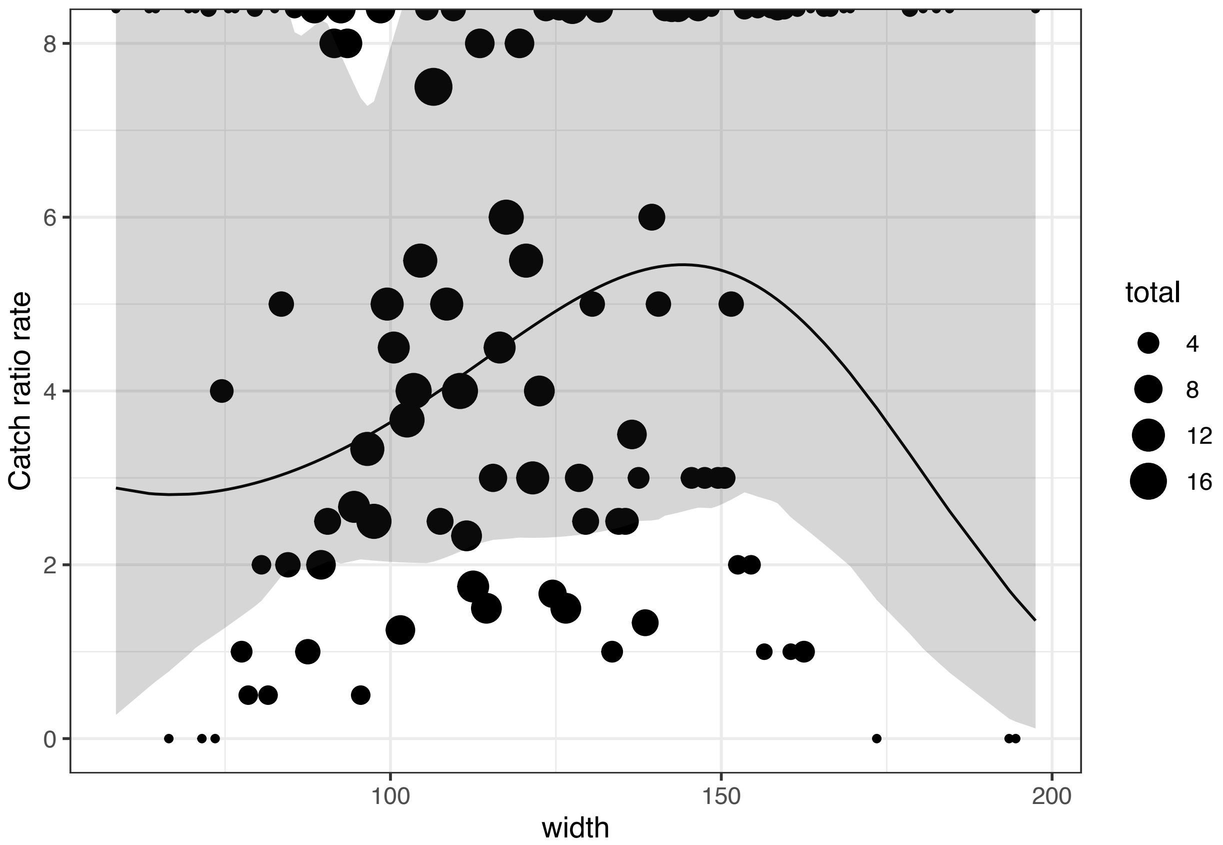
```
ggplot(sumdat, aes(width, prop))+geom_point(aes(size=total))+
   geom_line(data = newdata)+
   geom_ribbon(data = newdata, aes(ymin = lo, ymax = hi), alpha = 0.2)+
     ylab("Catch comparison rate")
```



A graphical comparison to the published results in Savina et al. (2017; in red) shows that the estimated catch comparison curves and relative CIs are very similar.



```
ggplot(sumdat, aes(width, prop/(1-prop))) + geom_point(aes(size = total))+
   geom_line(data = newdata)+
   geom_ribbon(data = newdata, aes(ymin = CRlo, ymax = CRhi), alpha = 0.2)+
    ylab("Catch ratio rate")+
    coord_cartesian(ylim = c(0,8))
```



The catch comparison curves properly reflected the trend in the experimental points. The experimental rates were subject to increasing binomial noise outside the length classes representing the main bulk of the catches. The results for edible crab showed significantly higher catches for 12 h at night compared to 12 h at day. On average, there were four times more catches for 12 h at night than 12 h at day. There was no strong indication of a length dependency in the data.

References

Herrmann, B., Sistiaga, M., Rindahl, L., Tatone, I. (2017). Estimation of the effect of gear design changes on catch efficiency: methodology and a case study for a Spanish longline fishery targeting hake (Merluccius merluccius). Fisheries Research, 185, 153-160.

Savina, E., Krag, L.A., Frandsen, R.P., and Madsen, N. (2017). Effect of fisher's soak tactic on catch pattern in the Danish gillnet plaice fishery. Fisehries Research, 196, 56-65.

Appendix 4: Catch comparison analyses of paired hauls of *Nephrops* twin-rigged trawls

7 Dec 2020

This example is based on the data from Melli et al. (2018). An anterior gear modification, namely the counter-herding device FLEXSELECT, was tested in a twin-rig configuration, where two identical bottom trawls were towed in parallel. One trawl was equipped with FLEXSELECT, while the other worked as baseline. The aim of the study was to determine if FLEXSELECT could prevent fish species from entering the trawl in a *Nephrops* (*Nephrops norvegicus*) fishery. Data for haddock *Melanogrammus aeglefinus* were collected for 21 hauls. Of these, 13 were conducted in day-time and 8 in night-time. In each haul and for each trawl, the total length (rounded down to the lower centimitre) of all haddock individuals was recorded.

Preliminaries

```
library(selfisher)
library(plyr) #for aggregating data across hauls
library(ggplot2); theme_set(theme_bw())
library(bbmle) #for AICtab
```

```
## Loading required package: stats4
```

```
library(stats4)
library(splines)
```

Data structure

First, we load the data and check out the variables. Here we can see that all the hauls are contained in one data frame (long format), with each row corresponding to a length class in a given haul. The number of individuals of that length-class caught in each trawl is reported in the columns `TEST1` (test trawl) and `TEST2` (baseline trawl). The column `TIME` classifies the haul as day-time (`D`) or night-time (`N`).

```
data("comphaddock")
head(comphaddock)
```

```
##   SPECIES HAUL TIME LENGTH TEST1 TEST2
## 1     Had   10    D    5.5     0     0
## 2     Had   10    D    6.5     0     0
## 3     Had   10    D    7.5     0     0
## 4     Had   10    D    8.5     0     0
## 5     Had   10    D    9.5     0     0
## 6     Had   10    D   10.5     0     1
```

To understand if the test trawl caught significantly fewer individuals of a given length-class than the baseline trawl we need to perform a catch comparison analysis (Krag et al., 2015). This analysis estimates the probability of catching an individual of a given length in the test trawl given that it was caught in either trawl.

In addition, the analysis aims at determining if the length-based efficiency of FLEXSELECT presents diel differences, as haddock is known to migrate vertically in the water column during the night.

Transforming data

Before fitting a model in `selfisher`, the following preparatory steps need to be performed:

1) Convert counts (i.e. number of individuals caught per length-class) to totals, proportions and ratios;

```
comphaddock = transform(comphaddock,
              total = TEST1 + TEST2,
              prop = TEST1 / (TEST1+TEST2),
              ratio = TEST1 / TEST2
)
```

This step is required because, unlike with other GLM functions for binomial regression, it is not possible in `selfisher` to specify the binomial variable as a two-column response variable, e.g. `cbind(N_TEST1, N_TEST2)`.

2) Remove eventual length classes where no individuals were caught;

```
comphaddock = subset(comphaddock, !is.na(prop))
```

3) Scale the length. This step is necessary for numerical stability, as a model often used for catch comparison analyses is the polynomial of order four, which requires to raise the length to the 4th power.

```
comphaddock$sl = scale(comphaddock$LENGTH)

head(comphaddock)
```

```
##    SPECIES HAUL TIME LENGTH TEST1 TEST2 total       prop     ratio        sl
## 6      Had   10    D   10.5     0     1     1 0.00000000 0.0000000 -1.750253
## 9      Had   10    D   13.5     0     5     5 0.00000000 0.0000000 -1.433650
## 10     Had   10    D   14.5     1    10    11 0.09090909 0.1000000 -1.328116
## 11     Had   10    D   15.5     8    15    23 0.34782609 0.5333333 -1.222581
## 12     Had   10    D   16.5    11    11    22 0.50000000 1.0000000 -1.117047
## 13     Had   10    D   17.5     7    21    28 0.25000000 0.3333333 -1.011513
```

Model fitting

The following is a typical model for catch comparison data with multiple paired hauls, which models the proportion of fish in the test versus the baseline trawl (`prop`) as a function of length (`sl`). This is epressed in the `selfisher` function by a two sided formula with the proportion (`prop`) on the left side and fixed and random effects on the right side. Formulas in the `selfisher` package follow the convention of the `lme4` and `glmmTMB` packages.

Since we are interested in determining if there is a length-dependent difference in the efficiency of the Test gear between day-time and night-time, we include in the model `TIME` as an explanatory variable.

```
m1 = selfisher(prop~(sl+I(sl^2)+I(sl^3)+I(sl^4))*TIME, total = total, comphaddock, haul = HAUL)
```

In this example all individuals were length-measured (i.e. there was no subsampling). In case of a subsampled species, an offset or q-ratio needs to be specified in the model. The `selfisher` function takes the `total` number of fish in the test and baseline using a separate argument, `total`. The argument `haul` needs to be specified in order to perform double-bootstrapping as demonstrated below. Otherwise, it could be omited from the model specificaiton as it doesn't affect the fit. The `haul` argument tells the software how to group the data for resampling in the bootstrapping procedure.

Alternative model

An alternative approach would consist in fitting a spline (Miller, 2013), often preferred to polynomial interpolation because it yields similar results while avoiding Runge's phenomenon (i.e. oscillation at the edges of the length range represented in the data).

```
m2 = selfisher(prop~(bs(sl, df=4))*TIME, total = total, comphaddock, haul=HAUL)
m3 = selfisher(prop~(bs(sl, df=5))*TIME, total = total, comphaddock, haul=HAUL)
```

Model comparison

We can determine which model fits best using the Akaike's Information Criterion (AIC; Akaike, 1974).

```
AICtab(m1, m2, m3)
```

```
##    dAIC df
## m3  0.0 12
## m1  0.4 10
## m2  6.8 10
```

This tells us that `m1` and `m3` show equally good fit (0.4 delta AIC units). According to the parsimony rule, we selected `m1` that is a simpler model.

Predictions

To see how the model fits the data, we need to plot observations and predictions together, keeping them separated by TIME.

```
dat_D = comphaddock[ which(comphaddock$TIME=='D'), ]
dat_N = comphaddock[ which(comphaddock$TIME=='N'), ]
newdata1 = data.frame(LENGTH = unique(dat_D$LENGTH), TIME = "D")
newdata2 = data.frame(LENGTH = unique(dat_N$LENGTH), TIME = "N")

newdata = rbind(newdata1, newdata2, deparse.level = 1)

newdata = transform(newdata,
                  sl = (LENGTH-mean(comphaddock$LENGTH))/sqrt(var(comphaddock$LENGTH)),
                  total = 1,
                  HAUL = NA
)

newdata$prop = predict(m1, newdata = newdata, type = "response")
```

Confidence intervals by double-bootstrapping

We then estimate the 95% Efron Confidence intervals (Efron, 1982), by accounting for within- and between-hauls variation (Millar, 1993). The code below resamples hauls, then resamples fish within hauls, fits the model to the resampled data, then makes predictions from the model onto `newdata`. The type of predictions we want in this case are the catch comparison rates, thus we specify `type="response"`. To read about the `predict` function, type `?predict.selfisher` in the R console.

Windows bootstrapping in parallel

```
library(snow)

ncpus = 4
cl = makeCluster(rep("localhost", ncpus))
clusterExport(cl, "newdata")
bs = bootSel(m1, nsim = 1000, parallel = "snow", cl = cl,
           FUN = function(mod){predict(mod, newdata = newdata, type = "response")})
stopCluster(cl)
```

Code for bootstrapping in Mac and Linux

```
bs = bootSel(m1, nsim = 1000, parallel = "multicore", ncpus = 4,
        FUN = function(mod){predict(mod, newdata = newdata, type = "response")})
```

Then we calculate quantiles across bootstraps for each row of `newdata`.

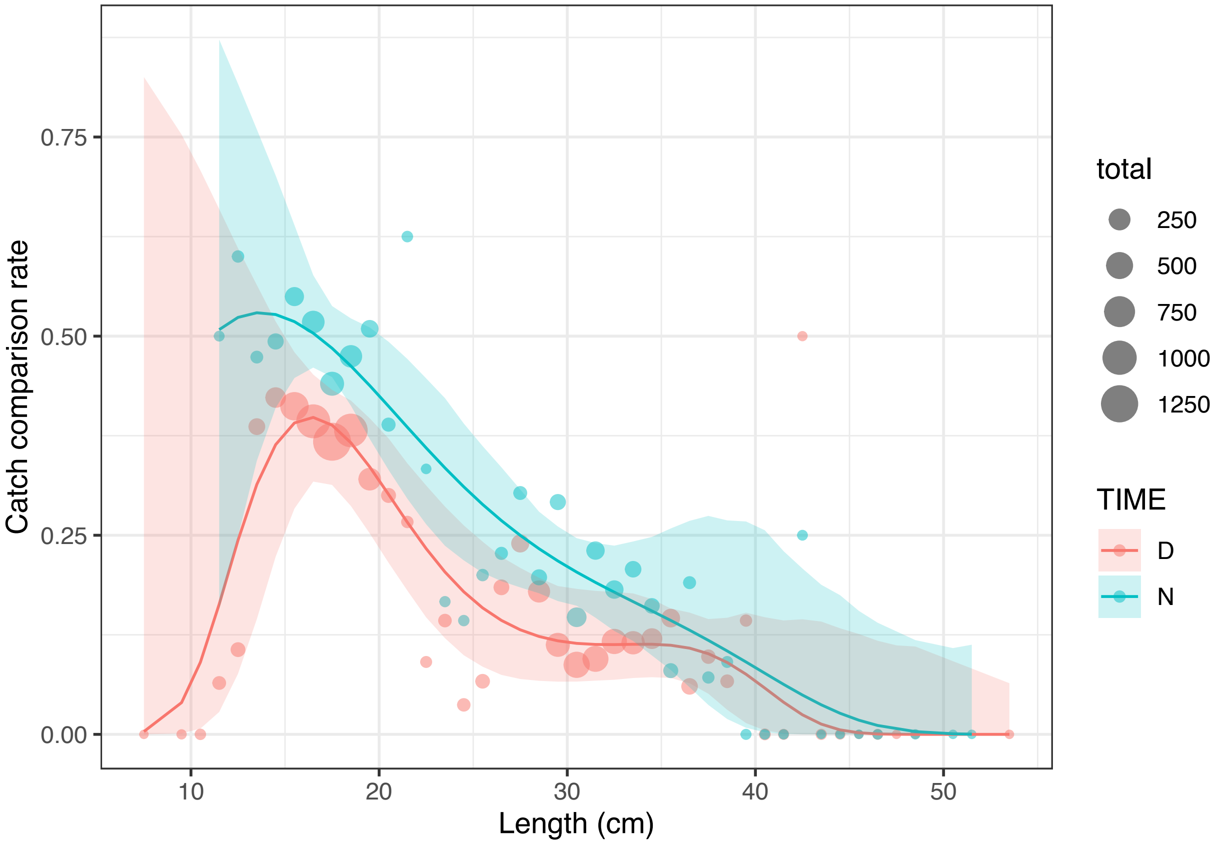
```
quants = apply(bs$t, 2, quantile, c(0.025, 0.5, 0.975))
newdata[,c("lo", "mid", "hi")] = t(quants)
```

Plotting with CIs

Here, we plot the modelled catch comparison curve with CIs and experimental obsevations, obtained by aggregating the hauls per TIME.

```
sumdat1 = ddply(dat_D, ~LENGTH+sl, summarize,
             prop = sum(TEST1)/sum(total),
             ratio = sum(TEST1)/sum(TEST2),
             total = sum(total),
             TIME = "D")

sumdat2 = ddply(dat_N, ~LENGTH+sl, summarize,
             prop = sum(TEST1)/sum(total),
             ratio = sum(TEST1)/sum(TEST2),
             total = sum(total),
             TIME = "N")

sumdat = rbind(sumdat1, sumdat2)
```

```
ggplot(sumdat, aes(LENGTH, prop))+geom_point(aes(size = total, col = TIME), alpha = 0.5)+
  geom_line(data = newdata, aes(col = TIME))+
  geom_ribbon(data = newdata, aes(ymin = lo, ymax = hi, fill = TIME), alpha = 0.2)+
  ylab("Catch comparison rate") + xlab("Length (cm)")
```



In accordance to Melli et al. (2018) a significant difference in catch comparison rate between day-time and night-time is found for individuals between 16 and 18 cm, as represented by the lack of overlapping between the CIs.

Melli et al. (2018) argued that, being the difference in a length range that is not usually retained when using a commercial codend, and that commercial fishing operations take place in both day- and night-time conditions, it is of greater interest to estimate the effect of FLEXSELECT with respect to the baseline trawl without the factor TIME.

Therefore, we repeat the steps of the process leaving out the factor TIME.

```
m4 = selfisher(prop~sl+I(sl^2)+I(sl^3)+I(sl^4), total = total, comphaddock, haul = HAUL)
m5 = selfisher(prop~bs(sl, df = 4), total = total, comphaddock, haul = HAUL)

AICtab(m4, m5)
```

```
##    dAIC df
## m4 0.0  5
## m5 5.1  5
```

Again, we select the polynomial of order 4 and use it to predict the catch comparison rates with CIs, which are then plotted against the overall experimental observations.

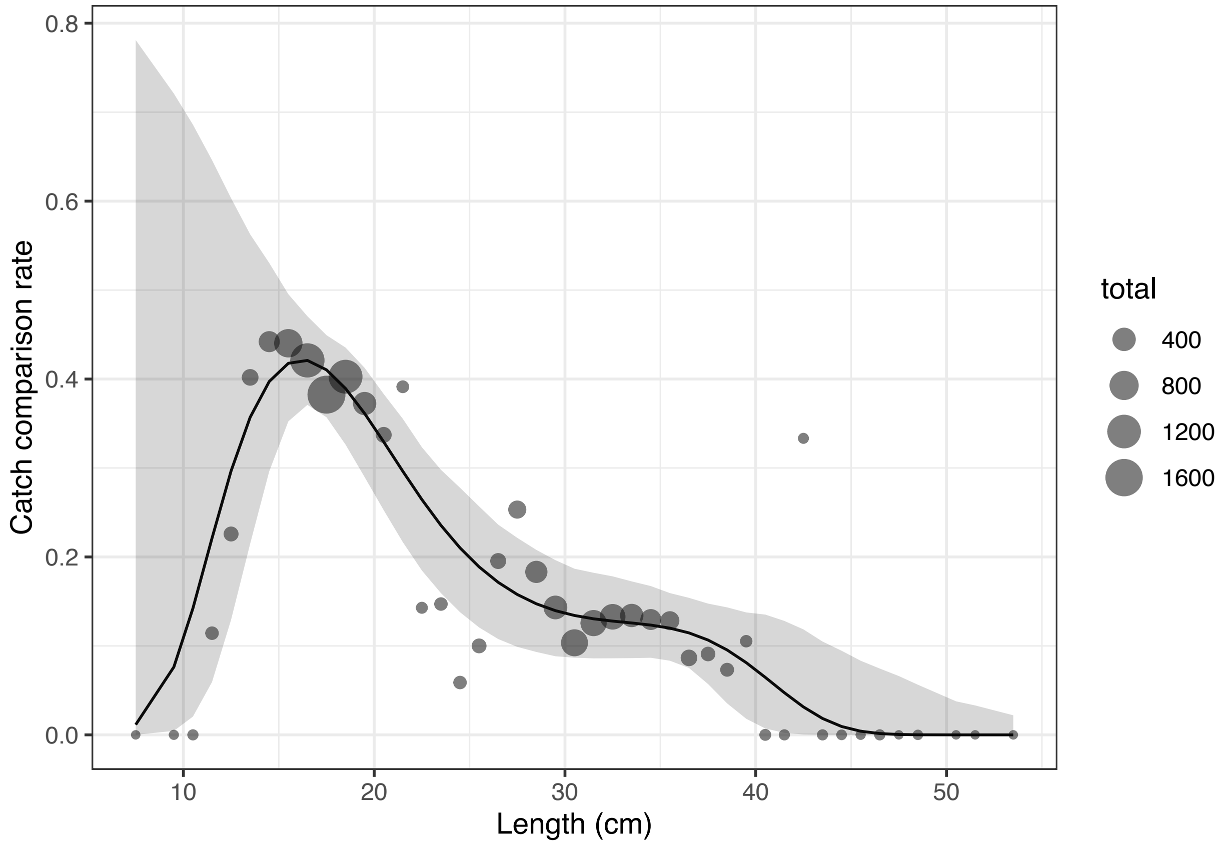
Windows code

```
ncpus = 4
cl = makeCluster(rep("localhost", ncpus))
clusterExport(cl, "newdata")
bs = bootSel(m4, nsim = 1000, parallel = "snow", cl = cl,
          FUN = function(mod){predict(mod, newdata = newdata, type = "response")})
stopCluster(cl)
```

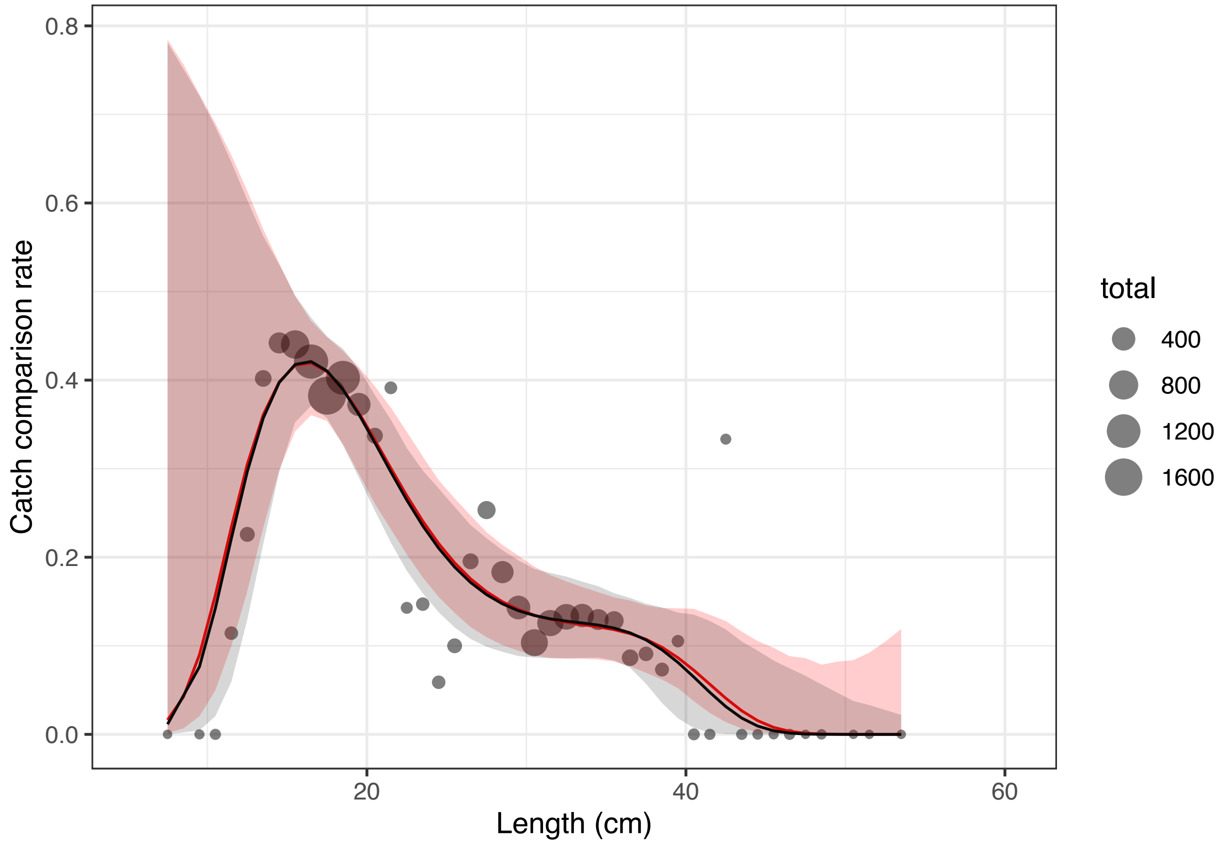
Mac and Linux code

```
bs = bootSel(m4, nsim = 1000, parallel = "multicore", ncpus = 4,
        FUN = function(mod){predict(mod, newdata = newdata, type = "response")})
```

```
quants = apply(bs$t, 2, quantile, c(0.025, 0.5, 0.975))
newdata[,c("lo", "mid", "hi")] = t(quants)

sumdat = ddply(comphaddock, ~LENGTH+sl, summarize,
             prop = sum(TEST1)/sum(total),
             ratio = sum(TEST1)/sum(TEST2),
             total = sum(total))

ggplot(sumdat, aes(LENGTH, prop))+geom_point(aes(size = total), alpha = 0.5)+
  geom_line(data = newdata)+
  geom_ribbon(data = newdata, aes(ymin = lo, ymax = hi), alpha = 0.2)+
  ylab("Catch comparison rate") + xlab("Length (cm)")
```



A graphical comparison to the published results in Melli et al. (2018; in red) shows that the estimated catch comparison curves and relative CIs are very similar.



Catch ratio

To directly quantify the difference in catch between the test and baseline trawls, it is common practice to estimate the catch ratio curve, using the relationship between catch comparison rate (cc) and catch ratio (cr): cr=cc/(1-cc)

First, we obtain predictions for the catch ratio specifying `type="ratio"` in the `predict` function.

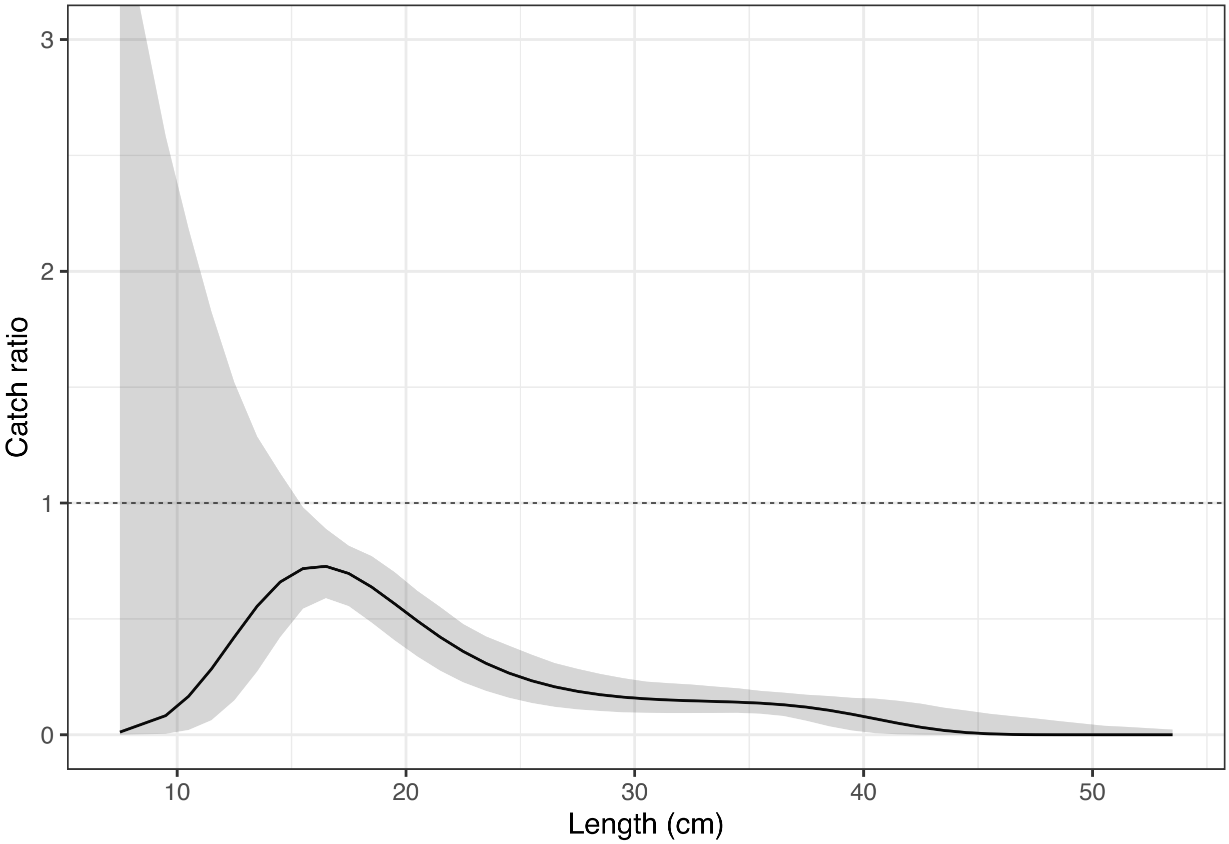
```
newdata$ratio = predict(m4, newdata = newdata, type = "ratio")
```

Second, we apply the relationship between cc and cr to obtain the CIs for the catch ratio curve.

```
bs$cr = bs$t/(1-bs$t)

CRquants = apply(bs$cr, 2, quantile, c(0.025, 0.5, 0.975))

newdata[,c("cr_lo", "cr_mid", "cr_hi")] = t(CRquants)
```

Catch ratio plot

```
ggplot(sumdat, aes(LENGTH, ratio))+
  geom_line(data = newdata, aes(y = ratio))+
  geom_hline(aes(yintercept = 1), size = 0.2, linetype = "dashed")+
  geom_ribbon(data = newdata, aes(ymin = cr_lo, ymax = cr_hi), alpha = 0.2)+
  ylab("Catch ratio")+ xlab("Length (cm)") +
  coord_cartesian(ylim = c(0, 3))
```



The results show that for individuals above 16 and up to 53 cm, the test gear with FLEXSELECT retained significantly fewer individuals. The effect is length-dependent, with a more pronounced reduction at larger length classes. Considering a minimum conservation reference size of 27cm for haddock in the fishing area (Skagerrak and Kattegat), the reduction of commercial-sized individuals is above 60%.

References

Akaike, H. (1974). A new look at the statistical model identification. IEEE T. Automat. Contr. 19, 716–722.

Efron, B. (1982). The jackknife, the bootstrap and other resampling plans. In Society for industrial and applied mathematics (SIAM) Monograph No. 38, CBSM-NSF.

Krag, L.A., Herrmann, B., Karlsen, J.D., and Mieske, B. (2015). Species selectivity in different sized topless trawl designs: Does size matter? Fish. Res. 172, 243-249.

Melli, V., Karlsen, J.D., Feekings, J.P., Herrmann, B., and Krag, L.A. (2018). FLEXSELECT: counter-herding device to reduce bycatch in crustacean trawl fisheries. Canad. J. Fish. Aquat. Sci. 75, 850–860.

Millar, R.B. (1993). Incorporation of between-haul variation using bootstrapping and nonparametric estimation of selection curves. Fish. Bullet. 91, 564–572.

Miller, T. (2013). A comparison of hierarchical models for relative catch efficiency based on paired-gear data for US Northwest Atlantic fish stocks. Can. J. Fish. Aquat. Sci. 70(9), 1306–1316.